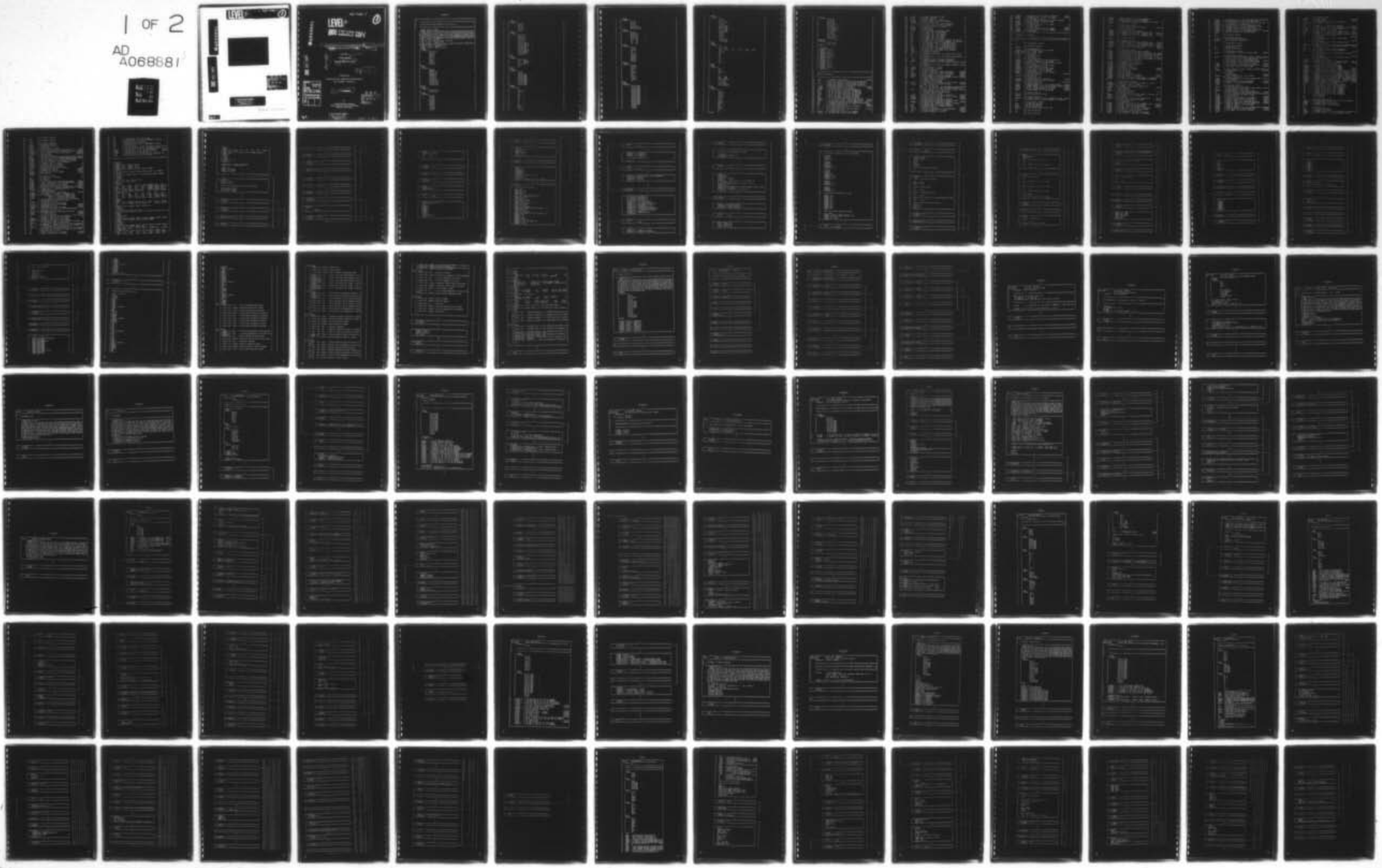
AD-A068 881   GENERAL ELECTRIC CO SYRACUSE N Y HEAVY MILITARY ELEC--ETC   F/G 17/1
SONAR SIMULATION COMPUTER PROGRAMS. VOLUME 2. FLOW CHARTS.(U)
NOV 68   N00140-68-C-0372

UNCLASSIFIED   NL

1 OF 2

AD
A068881



(6) SONAR SIMULATION COMPUTER PROGRAMS. Volume 2. Flow Charts.

PHASE I INTERIM TECHNICAL REPORT, on Phase 1.

(9)

VOLUME 2 B021 097L

FLOW CHARTS

Contract N00140-68-C-0372

(15)

001451

(12) 137p.

Prepared for

United States Navy Underwater Sound Laboratory
Fort Trumbull, Connecticut

(11) 21 Nov[redacted]68

By

General Electric Company
Heavy Military Electronics Systems
Syracuse, New York

```
                                    (ENTRANCE)
                                        I
                                        I
***************************************************************************
*C                                                                        *
*C   **********************************************************************
*C   *                                                                   **
*C   THIS IS MAIN CONTROL PROGRAM FOR SIMULATION                          *
*C   *                                                                   **
*C   **********************************************************************
*C                                                                        *
*     COMMON /TWOB2/ B2S,B2E                                              *
*     COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*    1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*    2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*    3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*    4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*    5LSUBE,ALSUBS,AAAAAA,BBBBBB,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*    6,FOS    , FRWS   , FRLOSE,                                          *
*    A PTS(128), FXS(128), PNS(128), FNS(128),                            *
*    B FOE    , FRWE   , FRLOEV, F2E                                      *
*    7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*    8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                    *
*     COMMON / ANGRAD / ANGBER(2)                                         *
*     COMMON / INTSPD / VXEINT, VYEINT                                    *
*     COMMON                                                              *
*    C / ARRAYC /                                                         *
*    D              NARRAY(2)                                             *
*    C              ARRYH1,                                               *
*    C              ARRYH2,                                               *
*    C              ARRYW1,                                               *
*    C              ARRYW2,                                               *
*    C              DELF   ,                                              *
*    C              FRES1  ,                                              *
*    C              FRES2  ,                                              *
*    C              QTRAN1,                                               *
*    C              QTRAN2,                                               *
*    NDUM2                                                                *
*     COMMON                                                              *
*    C / ARRAYP /                                                         *
*    C              DGANGS,                                               *
*    C              DGANGE,                                               *
*    D              ARRAYD(3,2)        ,                                  *
*    D              COSPHI(2)          ,                                  *
*    D              COSRAD(2)          ,                                  *
*    C              MSHIPS,                                               *
*    D              SINPHI(2)          ,                                  *
*    D              SINRAD(2)          ,                                  *
*    D              TARRIV(3,2)        ,                                  *
*    D              TMATSE(3,3)        ,                                  *
*    D              TMATEV(3,3)        ,                                  *
*    D              TVECTR(3,2)        ,                                  *
*    DDUMMYS                                                              *
*     COMMON                                                              *
*    C / BEAMCR /                                                         *
*    D              EULANG(3,4)                                           *
*    D              BEMCOR(3,4)                                           *
*     COMMON                                                              *
*    C / CALPHA /                                                         *
*    C              ALPHAE,                                               *
*    C              ALPHAS,                                               *
*    C              FALPHE,                                               *
*    C              FALPHS,                                               *
*    C              ONMIAE,                                               *
*    C              ONMIAS,                                               *
*    C              STNPEV,                                               *
*    C              STNPSE,                                               *
*    C              TWOALE,                                               *
*    C              TWOALS,                                               *
*    C              NTIMEM                                                *
```

```
  COMMON
 C / CONSIN /
 C              DEPSER,
 C              NCONSK,
 C              SPAISE,
 C              DEPEVA,
 C              NCONSL,
 C              SPAIEV
  COMMON
 C / FREQUN /
 D              COSAVE(2)           ,
 C              DFLRCZ,
 D              DYHITH(2)           ,
 D              ARAREA(2)           ,
 C              CTWOPI,
 D              SQAREA(2)           ,
 D              FRQSIG(128,2)       ,
 D              FRQLCN(128,2)       ,
 D              FRQIDB(128,2)       ,
 D              ANGDEP(2)           ,
 D              FRQTRN(128,2)       ,
 D              FRQASD(128,2)       ,
 D              FRQNCN(128,2)       ,
 D              FRQNOS(128,2)       ,
 D              NUMFRQ(2)           ,
 DDUM8
  COMMON
 C / INDEXS /
 D              ANDISO(16,2)      ,
 D              DI1    (50,16)      ,
 D              DI2    (50,16)      ,
 D              DLDIAN(16,2)        ,
 C              IO1    ,
 C              IO2    ,
 CDUMMYA
  COMMON
 C / LCONST /
 C              NULAPO,
 C              DEPROI,
 D              CONSG0(128)         ,
 D              CONSG1(128)         ,
 D              CONSG2(128)         ,
 D              CONSV0(128)         ,
 D              DELTAZ(128)         ,
 D              DEPKYD(128)         ,
 D              SLOPEJ(128)         ,
 D              SPDKYD(128)
  COMMON
 C / PTHLNG /
 C              A1     ,
 C              CI     ,
 C              CDSORD,
 C              CV     ,
 C              DI     ,
 C              DZ1    ,
 C              I      ,
 C              BI     ,
 C              SM     ,
 C              H      ,
 C              X      ,
 C              Y1     ,
 C              Y2     ,
 C              DXDC   ,
 C              PL     ,
 C              TIMCON
 COMMON
C / RANGES /
C               NUANMO,
C               ANGMAX,
C               DELANG,
C               DELRAD,
D               ANGINT(200)
D               RNGMOD(6,200)
 COMMON
C / RAYPAR /
C               RANGEH,
D               BOTANG(6)           .
D               DRDXDC(6)           .
D               PATHLN(6)           .
D               RANGEC(6)           .
D               SPI    (6)          .
D               SPT    (6)          .
D               TIR    (6)          .
D               TTR    (6)
 COMMON
C / RAYTRA /
C               NCONCI,
C               INITLK,
C               ZSTART,
C               Z2    ,
C               SPVRSQ,
C               ANGSTR,
C               ANGARR,
C               ANGBTM,
C               ANGSUR,
C               SPDVER,
C               RANGET
 COMMON
C / RCONST /
C               ZONACZ,
C               AMLSRD,
C               ZONBCZ,
C               HCI   ,
C               HZSD  ,
C               NCONSD,
C               RCZ1  ,
C               RCZ2  ,
C               SDCON ,
C               TCZAV1,
C               TCZAV2,
C               ZW    ,
NDUM3
 COMMON
C / SIGNAL /
D               PRYSEV(128)         .
D               PRYSSE(128)         .
D               PRNOEV(128)         .
D               PRNOSE(128)         .
D               PROEVA(128)         .
D               PROSER(128)         .
D               VAREVA(128)         .
D               VARSER(128)         .
D               GMUEVA(128)         .
D               GMUSER(128)         .
D               DEVAEV(128)         .
D               DEVASE(128)         .
D               DEMUEV(128)         .
D               DEMUSE(128)         .
C               THREVA,
C               THRSER,
C`              NTIMEN
```

```
 COMMON
C / STATIC /
C          NDSTAT,
C          NESTAT,
D          PEDEIN(128)
C          PERANG,
C          PPATHM,
D          PRANGE(128)     .
D          PRANGS(128)     .
D          PRPIEV(128)     .
D          PRPISE(128)     .
D          PSDEIN(128)     .
C          PSRANG,
C          SMPEDT,
C          SMPSDT,
C          SMTCNE,
C          SMTCNS,
D          TCONEN(128)     .
D          TCONSN(128)     .
CDUMMYB
 COMMON
C / STORAG /
D          ADI1  (16)      .
D          ADI2  (16)      .
C          AR1X  , AR1Y  , AR1Z  , AR2X  , AR2Y  , AR2Z  ,
C          ARRAY1,
C          ARRAY2,
C          E11   ,
C          E12   ,
C          E13   ,
C          E21   ,
C          E22   ,
C          E23   ,
C          F1E   ,
C          F1S   ,
CSDUMMY
 COMMON
C / SURDUC /
C          BLA1  ,
C          BLA2  ,
C          BLA3  ,
C          DIRAD ,
C          J     ,
C          M     ,
D          BF1   (128)     .
D          BF2   (128)     .
D          DELRAF(128,2)   .
D          FLN1(128)       .
D          FLN2(128)       .
D          CONLR2(40,50)   .
D          CONLR1(40,50)   .
DDUM9
 COMMON
C / SURFAC /
C          A6    ,
C          CONST2,
C          CONST4,
C          CZANGL,
C          CZANDL,
C          CZRANG,
C          G1SD  ,
C          G2SD  ,
C          NCZRAS,
C          NZONE ,
C          RSD   ,
C          RSD1  ,
C          SCSD  ,
C          SQRTZL,
C          SS    ,
C          ZL    ,
CDUMMYZ
```

```
*C                                                                              *
*       DIMENSION                                                               *
*      D              ANCZAV(2)         .                                       *
*      D              ARHITH(2)         .                                       *
*      D              ARQIRN(2)         .                                       *
*      D              ARWIDT(2)         .                                       *
*      D              BAFFUN(128,2)     .                                       *
*      D              CDYOVV(2)         .                                       *
*      D              DIRANG(16,2)      .                                       *
*      D              DIRSON(50,16,2)   .                                       *
*      D              FLONOS(128,2)     .                                       *
*      D              FRQLOW(2)         .                                       *
*      D              FRQRES(2)         .                                       *
*      D              NUDIAN(2)         .                                       *
*      D              SIGPLS(128,2)     .                                       *
*      D              SPDTER(2)         .                                       *
*      D              TMATRX(3,3,2)     .                                       *
*      DADUMMY(1)                                                               *
*       DIMENSION                                                               *
*      D              AINPUT(3,2)       .                                       *
*      D              EANGLE(3,2)       .                                       *
*      DDUMMYA(1)                                                               *
*C                                                                              *
*       EQUIVALENCE                                                             *
*      Q ( ANCZAV, TCZAV1 ),                                                    *
*      Q ( DIRANG, ADI1 ),                                                      *
*      Q ( ANGDGA, DGANGS ),                                                    *
*      Q ( ARHITH, ARRYH1 ),                                                    *
*      Q ( ARQTRN, QTRAN1 ),                                                    *
*      Q ( AINPUT, AR1X ),                                                      *
*      Q ( ARWIDT, ARRYW1 ),                                                    *
*      Q ( BAFFUN, BF1 ),                                                       *
*      Q ( DIRSON, DI1 ),                                                       *
*      Q ( EANGLE, E11 ),                                                       *
*      Q ( FLONOS, FLN1 ),                                                      *
*      Q ( FRQRES, FRES1 ),                                                     *
*      Q ( NUDIAN, I01 ),                                                       *
*      Q ( TMATRX, TMAISE ),                                                    *
*      Q ( NCONSR, NR )                                                         *
*C                                                                              *
*C      ************************************************************************
*C      *                                                                      **
*C      INPUTS TO PROGRAM                                                       *
*C      *                                                                      **
*C      ************************************************************************
*C                                                                              *
*C*     ACZ          = CONVERGENT ZONE CONSTANT                          (DIMENS*
*C*     ADI1   ( )   = ANGLES OF DIFFERENT DIRECTIVITIES FOR SEARCHER          (*
*C                   CHANGED FROM DEGREES TO RADIANS FOR INTERNAL USE           *
*C*     ADI2   ( )   = ANGLES OF DIFFERENT DIRECTIVITIES FOR EVADER            (*
*C                   CHANGED FROM DEGREES TO RADIANS FOR INTERNAL USE           *
*C*     ALPHAE       = SMOOTHING PARAMETER FOR EVADER                    (DIMENS*
*C*     ALPHAS       = SMOOTHING PARAMETER FOR SEARCHER                  (DIMENS*
*C*     ANGMAX       = MAXIMUM ANGLE BEING CONSIDERED FOR RAYS                 (*
*C*     AR1X         = ONE OF THE ARRAY DIMENSIONS FOR SEARCHER                 *
*C*     AR1Y         = ONE OF THE ARRAY DIMENSIONS FOR SEARCHER                 *
*C*     AR1Z         = ONE OF THE ARRAY DIMENSIONS FOR SEARCHER                 *
*C*     AR2X         = ONE OF THE ARRAY DIMENSIONS FOR EVADER                   *
*C*     AR2Y         = ONE OF THE ARRAY DIMENSIONS FOR EVADER                   *
*C*     AR2Z         = ONE OF THE ARRAY DIMENSIONS FOR EVADER                   *
*C*     ARRAY1       = TYPE OF ARRAY CONTROL CONSTANT FOR SEARCHER       (DIMENS*
*C*     ARRAY2       = TYPE OF ARRAY CONTROL CONSTANT FOR EVADER         (DIMENS*
*C*     B2E          = INTEGRATION BANDWIDTH OF POST-DETECTION OF EVADER        *
*C*     B2S          = INTEGRATION BANDWIDTH OF POST-DETECTION OF SEARCHER      *
*C*     BCZ          = CONVERGENT ZONE CONSTANT                          (DIMENS*
*C*     BF1    ( )   = BAFFLING CORRECTION FACTOR FOR SEARCHER           (DIMENS*
*C                   AS A FUNCTION OF ANGLE                                     *
*C*     BF2    ( )   = BAFFLING CORRECTION FACTOR FOR EVADER             (DIMENS*
*C                   AS A FUNCTION OF ANGLE                                     *
*C*     CONLR1( , )= RADIATED POWER SPECTRUM FROM SEARCHER                      *
*C*     CONLR2( , )= RADIATED POWER SPECTRUM FROM EVADER                        *
```

```
*C*     DELANG        = DIFFERENCE IN ANGLES FOR RAYS                       (*
*C*     DELF          = FREQUENCY INCREMENT                                  *
*C*     DELHED        = CHANGE IN HEADING OF EVADER                         (*
*C*     DELRNG        = CHANGE IN RANGE OF CLOSEST APPROACH                  *
*C*     DEPKYD( )     = DEPTH TO TOP OF LAYER                                *
*C*     DI1     ( , )= DIRECTIVITY INDEX FOR SONAR ON SEARCHER              *
*C*     DI2     ( , )= DIRECTIVITY INDEX FOR SONAR ON EVADER                *
*C*     DIFTI         = TIME BETWEEN POINTS ON A BRANCH                     (*
*C*     DPETEV        =  EVADER  DEPTH                                       *
*C*     DPETSE        = SEARCHER DEPTH                                       *
*C*     DTRAD         = ANGLE INCREMENT FOR RADIATED SIGNAL                 (*
*C*     E11           = EULER ANGLE INPUT FOR SEARCHER                      (*
*C*     E12           = EULER ANGLE INPUT FOR SEARCHER                       *
*C*     E13           = EULER ANGLE INPUT FOR SEARCHER                       *
*C*     E21           = EULER ANGLE INPUT FOR EVADER                         *
*C*     E22           = EULER ANGLE INPUT FOR EVADER                         *
*C*     E23           = EULER ANGLE INPUT FOR EVADER                         *
*C*     F0E           = PRE-DETECTION FILTER CENTER FREQUENCY FOR EVADER     *
*C*     F0S           = PRE-DETECTION FILTER CENTER FREQUENCY FOR SEARCHER   *
*C*     F1E           = LOWER FREQUENCY LIMIT OF EVADER EQUIPMENT            *
*C*     F1S           = LOWER FREQUENCY LIMIT OF SEARCHER EQUIPMENT          *
*C*     F2E           = UPPER FREQUENCY LIMIT OF EVADER EQUIPMENT            *
*C*     F2S           = UPPER FREQUENCY LIMIT OF SEARCHER EQUIPMENT          *
*C*     FBWE          = PRE-DETECTION FILTER BANDWIDTH OF EVADER             *
*C*     FBWS          = PRE-DETECTION FILTER BANDWIDTH OF SEARCHER           *
*C*     FLN1    ( )   = FLOW NOISE OF SEARCHER                               *
*C*     FLN2    ( )   = FLOW NOISE OF EVADER                                 *
*C      FNE     ( )   = ORDERED FREQUENCY POINTS FOR NOISE AROUND EVADER     *
*C                      SAME AS FXE                                          *
*C      FNS     ( )   = ORDERED FREQUENCY POINTS FOR NOISE AROUND SEARCHER   *
*C                      SAME AS FXS                                          *
*C*     FRES1         = TRANSDUCER RESONANT FREQUENCY ON SEARCHER            *
*C*     FRES2         = TRANSDUCER RESONANT FREQUENCY ON EVADER              *
*C      FXE     ( )   = ORDERED FREQUENCY POINTS FOR SIGNAL RECIEVED BY EVADE*
*C                      SAME AS FNE                                          *
*C      FXS     ( )   = ORDERED FREQUENCY POINTS FOR SIGNAL RECIEVED BY SEARC*
*C                      SAME AS FNS                                          *
*C*     HDINEV        = INITIAL HEADING OF EVADER                           (*
*C*     HEDMAX        = MAXIMUM HEADING OF EVADER                           (*
*C*     HS1           = HEADING OF SEARCHER                                 (*
*C*     I01           = NUMBER OF DIRECTIVITY ANGLES FOR SEARCH       (DIMENS*
*C*     I02           = NUMBER OF DIRECTIVITY ANGLES FOR EVADER       (DIMENS*
*C*     MAXLAY        = MAXIMUM LAYERS TO BE USED TO FI7 PROFILE      (DIMENS*
*C*     MAXTIM        = MAXIMUM NUMBER OF POINTS ON A BRANCH          (DIMENS*
*C*     MECO          = CONTROL PARAMETER FOR EVASION COURSE OPTIONS  (DIMENS*
*C                      1 = INVERSE BEARING RIDER OPTION                     *
*C                      2 = NORMAL TO SEARCH EVASION                         *
*C                      3 = CONTINUE INITIAL COURSE                          *
*C*     NEMAX         = MAXIMUM POINT ALONG BRANCH TO START EVASION   (DIMENS*
*C*     NEWLAY        = CONTROL CONSTANT FOR INITIALIZATION           (DIMENS*
*C                      NEGATIVE = NEW PROFILE                               *
*C                      ZERO      = OLD PROFILE-- NEW SHIP DEPTHS OR DELTA ANGLE*
*C                      POSITIVE = OLD PROFILE-- OLD SHIP DEPTHS AND ANGLES  *
*C*     NPCO          = CONTROL PARAMETER FOR PURSUIT COURSE OPTIONS  (DIMENS*
*C                      1 = BEARING RIDER CLOSING TACTICS                    *
*C                      2 = COLLISION COURSE CLOSING TACTICS                 *
*C                      3 = CONTINUE INITIAL COURSE                          *
*C*     NPRINT        = CONTROL PARAMETER FOR AMOUNT OF PRINTOUT      (DIMENS*
*C*     NSMAX         = MAXIMUM POINT ALONG BRANCH TO START CLOSING   (DIMENS*
*C*     NULAPO        = NUMBER OF LAYERS PLUS ONE                     (DIMENS*
*C*     PDEMIN        = MINIMUM DETECTION PROBABILITY ALLOWED         (DIMENS*
*C                      AFTER SHIPS START SEPARATING AS SEEN BY EVADER       *
*C*     PDSMIN        = MINIMUM DETECTION PROBABILITY ALLOWED         (DIMENS*
*C        .             AFTER SHIPS START SEPARATING AS SEEN BY SEARCHER     *
*C*     PHIE          = DEPRESSION ANGLE FOR EVADER STEERING                (*
*C*     PHIS          = DEPRESSION ANGLE FOR SEARCHER STEERING              (*
*C*     POR           = POROSITY OF BOTTOM                            (DIMENS*
*C*     PPAMIN        = MINIMUM PATH PROBABILITY TO BE CONSIDERED     (DIMENS*
*C*     PRE           = A-PRIORI PROBABILITY OF EVASION               (DIMENS*
*C*     PRK           = A-PRIORI PROBABILITY OF A KILL                (DIMENS*
*C*     QTRAN1      = TRANSDUCER FIGURE OF MERIT ON SEARCHER         (DIMENS*
*C*     QTRAN2      = TRANSDUCER FIGURE OF MERIT ON EVADER           (DIMENS*
*C*     RGINEV      = INITIAL RANGE OF CLOSE                                *
*C*     RI          = INITIAL RANGE BETWEEN SHIPS AT START OF BRANCH        *
*C*     RNGMAX      = MAXIMUM CLOSEST APPROACH DISTANCE                     *
*C*     SPDEVA      = SPEED OF EVADER                                       *
*C*     SPDKYD( )   = PROPAGATION SPEED AT TOP OF LAYER                   (K*
*C*     SPDSER      = SPEED OF SEARCHER                                     *
*C*     SS          = SEA STATE INPUT                                (DIMENS*
*C*     THREVA      = DETECTION THRESHOLD FOR THE EVADER             (DIMENS*
*C*     THRSER      = DETECTION THRESHOLD FOR THE SEARCHER           (DIMENS*
*C*     WRANGE      = WEAPON RANGE                                          *
*C  *********************************************************************
*C  *                                                                   *
*C *   CONSTANTS  IN  PROGRAM                                            *
*C  *                                                                   *
*C  *********************************************************************
*C  *********************************************************************
*C  *                                                                   *
*C      A1          =                                                   *
*C                  = COMPUTATIONAL VARIABLE                            *
*C      A6          =                                                   *
*C                  COMPUTATIONAL VARIABLE                              *
*C      AAAAAA      =                                                   *
*C                  NOT USED OR REQUIRED                                *
*C      AINPUT( , )= ARRAY PARAMETERS INPUT TO PROGRAM                  *
*C *    ALPXN = VALUE GEN. IN SUB. UPDATE                               *
*C *    ALPYN = VALUE GEN. IN SUB. UPDATE                               *
*C      ALSUBE      = -SEE ALPHAE-                                      *
*C                  NOT USED OR REQUIRED                                *
*C      ALSUBS      = -SEE ALPHAS-                                      *
*C                  NOT USED OR REQUIRED                                *
*C      AMLSRD      = CONSTANT REQQUIRED FOR AMOS CALCULATIONS          *
*C      ANCZAV( )   = AVERAGE OF CONVERGENT ZONE ANGLES                (*
*C      ANDISO( , )= ANGLES OF DIFFERENT DIRECTIVITIES FOR SHIPS       (*
*C      ANGARR      = ARRIVAL ANGLE OF RAY AT Z2                       (*
*C      ANGBTM      = BOTTOM BOUNCE ANGLE OF RAY PORTION               (*
*C      ANGDEP( )   = STEERING DEPRESSION ANGLES                        *
*C      ANGDGA( )   = AZIMUTHAL MAIN BEAM STEERING ANGLE               (*
*C      ANGINT( )   = INITIAL ANGLE OF RAY AT SEARCHER                 (*
*C      ANGSTR      = STARTING ANGLE OF RAY AT ZSTART                  (*
*C      ANGSUR      = SURFACE BOUNCE ANGLE OF RAY PORTION              (*
*C      ARAREA( )   = ARRAY AREA CONSTANT                      ((YD-SEC/K*
*C                  NOT USED OR REQUIRED                                *
*C      ARHITH( )   = ARRAY HEIGTH                                      *
*C                  NOT USED OR REQUIRED                                *
*C      ARQTRN( )   = Q-FACTOR FOR ARRAY TRANSDUCERS              (DIMENS*
*C      ARRAYD( , )= MODIFIED ARRAY DIMENSIONS FOR SHIPS                *
*C      ARWIDT( )   = ARRAY WIDTH CONSTANT                     ((YD.SEC/K*
*C                  NOT USED OR REQUIRED                                *
*C      AVBRKL      = AVERAGE PROBABILITY OF KILL ALONG A BRANCH   (DIMENS*
*C      AVBREV      = AVERAGE PROBABILITY OF EVASION ALONG A BRANCH (DIMENS*
*C      AVEEVA      = OVERALL AVERAGE PROBABILITY OF EVADE         (DIMENS*
*C      AVEKIL      = OVERALL AVERAGE PROBABILITY OF KILL          (DIMENS*
*C      AVEPTH      = AVERAGE PROBABILITY OF CONTINUING            (DIMENS*
*C      B2          =                                                   *
*C                  NOT USED OR REQUIRED                                *
*C      BAFFUN( , )= BAFFLING VALUES FOR SHIPS                          *
*C      BBBBBB      =                                                   *
*C                  NOT USED OR REQUIRED                                *
*C      BEMCOR( , )= FREQUENCY INDEPENDENT BEAM-CORRECTION FACTOR  (DIMENS*
*C      BFP         = AZIMUTHAL ANGLE OF EVADER MEASURED FROM SEARCHER  (*
*C      B1          =                                                   *
*C                  = COMPUTATIONAL VARIABLE                            *
*C      BLA1        = BOTTOM LOSS CONSTANT                              *
*C      BLA2        = BOTTOM LOSS CONSTANT                              *
*C      BLA3        = BOTTOM LOSS CONSTANT                              *
*C      BOTANG( )   = BOTTOM BOUNCE ANGLE FOR RAY TYPE                 (*
*C      BPE         =                                                   *
*C                  NOT USED OR REQUIRED                                *
*C      BPS         =                                                   *
*C                  NOT USED OR REQUIRED                                *
```

```
*C      BRNEVA       = BRANCH SUMMATION OF EVASION PROBABILITY                    *
*C      BRNKIL       = BRANCH SUMMATION OF KILL PROBABILITY                       *
*C      BSP          = AZIMUTHAL ANGLE OF SEARCHER MEASURED FROM EVADER         (*
*C      CDSQRD       = COMPUTATIONAL VARIABLE                                     *
*C      CI           =                                                            *
*C                   = COMPUTATIONAL VARIABLE                                     *
*C      CLPH         = CONTROL CONSTANT FOR CLOSE PHASE                   (DIMENS*
*C                   0 = NOT IN CLOSE PHASE                                       *
*C                   1 = IN CLOSE PHASE                                           *
*C      CONNOR       = CONSTANT USED FOR NORMALIZING                              *
*C      CONSG0( )    = G0 CONSTANT AS COMPUTED BY LAYERS ROUTINE         (SEC**2/*
*C      CONSG1( )    = G1 CONSTANT AS COMPUTED BY LAYERS ROUTINE         (SEC**2/*
*C      CONSG2( )    = G2 CONSTANT AS COMPUTED BY LAYERS ROUTINE               (K*
*C      CONST2       =                                                            *
*C                   COMPUTATIONAL VARIABLE                                       *
*C      CONST4       =                                                            *
*C                   COMPUTATIONAL VARIABLE                                       *
*C      CONSV0( )    = V0 CONSTANT AS COMPUTED BY LAYERS ROUTINE         ((SEC/K*
*C      COSAVE( )    = COSINE OF AVERAGE OF CONVERGENT ZONE ANGLES       (DIMENS*
*C      COSPHI( )    = COSINE OF STEERING DEPRESSION ANGLE FOR SHIPS (DIMENS*
*C      COSRAD( )    = COSINE OF AZIMUTHAL SIGNAL ARRIVAL ANGLE          (DIMENS*
*C      CTWOPI       = CONSTANT EQUAL TO TWO TIMES PI                    (DIMENS*
*C      CV           =                                                            *
*C                   = COMPUTATIONAL VARIABLE                                     *
*C      CZANDL       = DELTA ANGLE FOR CONVERGENT ZONE SEARCH                   (*
*C      CZANEO       = SMALLER CONVERGENT ZONE ANGLE (IN RADIANS) FOR EVADER*
*C      CZANET       = LARGER  CONVERGENT ZONE ANGLE (IN RADIANS) FOR EVADER*
*C      CZANGL       = ANGLE BEING USED FOR CONVERGENT SONE RANGE SEARCH    (*
*C      CZANSO       = SMALLER CONVERGENT ZONE ANGLE (IN RADIANS) FOR SEARCH*
*C      CZANST       = LARGER  CONVERGENT ZONE ANGLE (IN RADIANS) FOR SEARCH*
*C      CZRANG       = RANGE FOR CONVERGENT ZONE RAY                              *
*C      D            = DEPTH (IN K-YD) IN LAYER                                   *
*C      DELBAF( , )= DIFFERENCE IN BAFFLING VALUES                                *
*C      DELRAD       = DIFFERENCE IN ANGLES FOR RAYS                            (*
*C      DELRCZ       = WIDTH OF CONVERGENT ZONE                                   *
*C      DELTAZ( )    = THICKNESS OF LAYER                                         *
*C      DEMUEV( )    = MODIFIED MU FOR EVADER                            (DIMENS*
*C      DEMUSE( )    = MODIFIED MU FOR SEARCHER                          (DIMENS*
*C      DEPBOT       = DEPTH OF BOTTOM                                            *
*C      DEPEVA       = DEPTH OF EVADER                                            *
*C      DEPSER       = DEPTH OF SEARCHER                                          *
*C      DEVAEV( )    = MODIFIED VARIANCE FOR EVADER                      (DIMENS*
*C      DEVASE( )    = MODIFIED VARIANCE FOR SEARCHER                    (DIMENS*
*C      DGANGS       = AZIMUTHAL MAIN-BEAM STEERING ANGLE FOR SEARCHER      (*
*C      DGANGE       = AZIMUTHAL MAIN-BEAM STEERING ANGLE FOR SEARCHER      (*
*C      DI           =                                                            *
*C                   = COMPUTATIONAL VARIABLE                                     *
*C      DIRANG( , )= ANGLES USED WITH DIFFERENT DIRECTIVITY CURVES             (*
*C      DIRSON(,,) = DIRECTIVITY INDEX FOR SONAR                                  *
*C      DLDIAN( , )= DIFFERENCES IN ANGLES OF DIRECTIVITIES                    (*
*C      DRDMDC( )    = DERIVATIVE (DXDC) FOR RAY TYPE                    ((SEC/K*
*C      DUMMY1       = DUMMY VARIABLE                                             *
*C      DXDC         = RANGE DERIVATIVE                                           *
*C      DYHITH( )    = ARRAY HEIGTH CONSTANT                                (YD-S*
*C                   NOT USED OR REQUIRED                                         *
*C      DZ1          = DEPTH (IN K-YD) FROM TOP OF LAYER TO STARTING POINT (*
*C      EANGLE( , )= EULER ANGLES INPUT TO PROGRAM                                *
*C      EDEPTH       = SEE -DEPEVA-                                               *
*C      EULANG( , )= EULERIAN ANGLE AND THEIR'SINE AND COSINE            (DIMENS*
*C      EVPH         = CONTROL CONSTANT FOR EVADE PHASE                  (DIMENS*
*C                   0 = NOT IN EVADE PHASE                                       *
*C                   1 = IN EVADE PHASE                                           *
*C      FACTOR       = MODIFICATION FACTOR FOR PATH PROBABILITIES        (DIMENS*
*C      FALPHE       = FRACTION  (ONMIAE**2/(1-ONMIAE**2))               (DIMENS*
*C      FALPHS       = FRACTION  (ONMIAS**2/(1-ONMIAS**2))               (DIMENS*
*C      FILSER       = FUNCTION FOR FILTER RESPONSE OF SEARCHER                   *
*C      FILEVA       = FUNCTION FOR FILTER RESPONSE OF EVADER                     *
*C      FLONOS( , )= FLOW AND SELF NOISE FOR SHIPS                                *
*C      FRLOEV       = LOWER INTEGRATION FREQUENCY FOR  EVADER                    *
*C      FRLOSE       = LOWER INTEGRATION FREQUENCY FOR SEARCHER                   *
```

```
*C      FRQASD( , )= INTER-RANGE CONSTANT TIME SQRT( FREQUENCY/LAYER DEPTH*
*C      FRQIDB( , )= TEN.LOG( FREQUENCY ) AS A FUNCTION OF SHIP            *
*C      FRQLCN( , )= LAYER CONSTANT AS A FUNCTION OF FREQUENCY AND SHIP    *
*C      FRQLOW( )  = LOWER FREQUENCY OF INTEGRATION                        *
*C      FRQNCN( , )= NOISE SPECTRA UNALTERED BY SONAR DIRECTIVITY          *
*C      FRQNOS( , )= FREQUENCY POINTS OF NOISE  AS FUNCTION OF SHIP        *
*C      FRQRES( )  = TRANSDUCER RESONANT FREQUENCY FOR SHIPS               *
*C      FRQSIG( , )= FREQUENCY POINTS USED FOR SIGNAL COMPUTATIONS         *
*C      FRQTRN( , )= FREQUENCY RESPONSE OF TRANSDUCER ELEMENTS     (DIMENS*
*C      FTTOKY     = FUNCTION TO CHANGE FEET TO KILO-YARDS                 *
*C      G1SD       =                                                       *
*C                 COMPUTATIONAL VARIABLE                                  *
*C      G2SD       =                                                       *
*C                 COMPUTATIONAL VARIABLE                                  *
*C      GMUSER( )  = MEAN OF SMOOTHED SEARCHER S/N                         *
*C      GMUEVA( )  = MEAN OF SMOOTHED EVADER S/N                           *
*C      H          =                                                       *
*C                 = COMPUTATIONAL VARIABLE                                *
*C      HCI        =                                                       *
*C                 = COMPUTATIONAL VARIABLE                                *
*C      HE1        = HEADING OF EVADER                                     (*
*C      HZSD       = SURFACE DUCT CONSTANT                                 *
*C      I          =                                                       *
*C                 = COMPUTATIONAL VARIABLE                                *
*C      INITLK     = LAYER NUMBER IN WHICH RAY PORTION STARTS      (DIMENS*
*C      J          =                                                       *
*C                 = COMPUTATIONAL VARIABLE                                *
*C      JUMPIF     = CONTROL PARAMETER FOR RECALCULATING A LAYER CONSTANT *
*C      K          = INITIALLY THE LAYER IN WHICH THE STARTING POINT IS FO*
*C                 LAYER IN WHICH CALCULATIONS ARE BEING MADE              *
*C      K          = INDEX VARIABLE USED IN INTEGRATION ROUTINE    (DIMENS*
*C      L          = LAYER IN WHICH FINAL POINT (Z2) IS LOCATED            *
*C      LAYERL     = LAYER IN WHICH A LOWER VERTEX IS FOUND        (DIMENS*
*C      LAYERM     = LAYER IN WHICH VELOCITY PROFILE HAS A ZERO    (DIMENS*
*C      LAYERS     = LAYER WHICH CONTAINS SHALLOWER SHIP           (DIMENS*
*C      M          =                                                       *
*C      MAXLAY     = MAXIMUM NUMBER OF LAYERS ALLOWED INCLUDING ADDED POIN*
*C      MSHIPS     = CONTROL NUMBER OF SHIP                        (DIMENS*
*C                 1 = SEARCHER                                            *
*C                 2 = EVADER                                              *
*C      N          = POINT ON BRANCH                               (DIMENS*
*C      NARRAY( )  = CONTROL PARAMETER FOR TYPE OF ARRAY           (DIMENS*
*C                 NOT USED OR REQUIRED                                    *
*C      NCONCI     = CONTROL PARAMETER FOR DIFFERENT SUBROUTINES TO BE CAL*
*C      NCONSD     = AMOS CONSTANT SD                                      *
*C      NCONSK   * = NUMBER OF LAYER CONTAINING SEARCHER           (DIMENS*
*C      NCONSL     = NUMBER OF LAYER CONTAINING EVADER             (DIMENS*
*C      NCZRAS     =                                                       *
*C                 COMPUTATIONAL VARIABLE                                  *
*C      NDSTAT     = INTEGER VALUE OF THE D-STATE                          *
*C      NEI        = POINT ALONG BRANCH WHEN EVADER STARTS EVADING (DIMENS*
*C      NESTAT     = INTEGER VALUE OF THE E-STATE                          *
*C      NNFLAG     = DEBUG PARAMETER TO SHOW PROGRAM FLOW                  *
*C      NR         = CONTROL CONSTANT FOR POSITION IN BRANCHES     (DIMENS*
*C      NSI        = POINT ALONG BRANCH WHEN SEARCHER START CLOSING(DIMENS*
*C      NSTART     = CONTROL PARAMETER FOR START OF LOOP                   *
*C      NTIMEM     = NTIMEN MINUS ONE                              (DIMENS*
*C      NTIMEN     = POSITION ALONG PROBABILITY TREE BRANCH        (DIMENS*
*C      NUANMO     = NUMBER OF ANGLES FOR RAYS MINUS ONE           (DIMENS*
*C      NUDIAN( )  = NUMBER OF DIRECTIVITY ANGLES FOR SHIPS        (DIMENS*
*C      NUMANG     = NUMBER OF ANGLE BEING CONSIDERED FOR RAYS     (DIMENS*
*C      NUMFRQ( )  = NUMBER OF FREQUENCY POINTS FOR INTEGRATION    (DIMENS*
*C      NUMHED     = NUMBER OF HEADINGS TO BE CONSIDERED                   *
*C      NUMLAY     = NUMBER OF LAYERS                                      *
*C      NUMRNG     = NUMBER OF RANGES FOR CLOSEST APPROACH BEING CONSIDERE*
*C      NVTXLO     = LAYER NUMBER BELOW LOWER VERTEX LAYER         (DIMENS*
*C      NVTXUP     = LAYER NUMBER IN WHICH RAY VERTEXES            (DIMENS*
*C      NZONE      =                                                       *
```

```
*C                  COMPUTATIONAL VARIABLE                                       *
*C      ONMIAE    = ONE MINUS ALPHAE                                  (DIMENS*
*C      ONMIAS    = ONE MINUS ALPHAS                                  (DIMENS*
*C      P         = DUMMY VARIABLE FOR COMPUTING PATHLENGTHS                     *
*C      PATHLN( ) = PATH LENGTH OF RAY TYPE                                      *
*C      PDE    ( ) =                                                             *
*C      PDS    ( ) = PROBABILITY OF DETECTION BY SEARCHER             (DIMENS*
*C      PE     ( ) = PROBABILITY OF EVASION AS A FUNCTION OF STATE (DIMENS*
*C      PEDETN( ) = INITIAL PROBABILITY OF  EVADER DETECTION          (DIMENS*
*C      PERANG    = RANGE OF DETECTION BY EVADER                                 *
*C      PEVADE( ) = PROBABILITY OF EVASION AT EACH BRANCH POINT      (DIMENS*
*C      PGE    ( ) = PROBABILITY OF SEARCHER EVENT OTHER THAN KILL (DIMENS*
*C      PGS    ( ) = PROBABILITY OF EVADER EVENT OTHER THAN EVADE   (DIMENS*
*C      PHI       = ANGLE RAY MAKES WITH SHIP AXIS                             (*
*C      PIE    ( ) = SEE -PRYSEV-                                     (DIMENS*
*C                  A-PRIORI PROBABILITY OF DETECTION BY EVADER                  *
*C                  NOT USED OR REQUIRED                                         *
*C      PIS    ( ) = SEE -PRYSSE-                                     (DIMENS*
*C                  A-PRIORI PROBABILITY OF DETECTION BY SEARCHER                *
*C                  NOT USED OR REQUIRED                                         *
*C      PKDS   ( ) = PROBABILITY OF KILL GIVEN DETECTION BY SEARCHE(DIMENS*
*C      PKILL  ( ) = PROBABILITY OF KILL AT EACH POINT ON BRANCH     (DIMENS*
*C      P_L       = PATH LENGTH OF RAY                                           *
*C      PNE    ( ) = NOISE POWER SPECTRUM AS SEEN BY EVADER                      *
*C      PNS    ( ) = NOISE POWER SPECTRUM AS SEEN BY SEARCHER                    *
*C      POE    ( ) = SEE -PRNOEV-                                     (DIMENS*
*C                  A-PRIORI PROBABILITY THAT NO DETECTION BY EVADER             *
*C                  NOT USED OR REQUIRED                                         *
*C      POS    ( ) = SEE -PRNOSE-                                     (DIMENS*
*C                  A-PRIORI PROBABILITY OF NO DETECTION BY SEARCHER             *
*C                  NOT USED OR REQUIRED                                         *
*C      POWRCD    = COMPUTATIONAL VARIABLE                                       *
*C      PPATH ( ) = PROBABILITY THAT POINT ON BRANCH IS PASSED       (DIMENS*
*C      PPATHM    = PROBABILITY OF REACHING THE PRESENT POINT         (DIMENS*
*C                  I. E. P-OF-PATH FOR N-MINUS-ONE                              *
*C      PRANGE( ) = RANGE OF FIRST DETECTION BY EVADER                           *
*C      PRANGS( ) = RANGE OF FIRST DETECTION BY SEARCHER                         *
*C      PROEVA( ) =  PROBABILITY OF DETECTION BY EVADER               (DIMENS*
*C      PROSER( ) =  PROBABILITY OF DETECTION BY SEARCHER             (DIMENS*
*C      PRNOEV( ) =  EVADER  DECISION PROBABILITY OF NO-DETECTION (DIMENS*
*C      PRNOSE( ) = SEARCHER DECISION PROBABILITY OF NO-DETECTION (DIMENS*
*C      PRNTEV    = PROBABILITY OF NOT BEING DETECTED BY  EVADER   (DIMENS*
*C      PRNTSE    = PROBABILITY OF NOT BEING DETECTED BY SEARCHER (DIMENS*
*C      PRPTEV( ) = PROBABILITY OF PATH  OF DETECTIONS FOR  EVADER(DIMENS*
*C      PRPTSE( ) = PROBABILITY OF PATH  OF DETECTIONS FOR SEARCHE(DIMENS*
*C      PRYSEV( ) =  EVADER  DECISION PROBABILITY OF    DETECTION (DIMENS*
*C      PRYSSE( ) = SEARCHER DECISION PROBABILITY OF    DETECTION (DIMENS*
*C      PSDETN( ) = INITIAL PROBABILITY OF SEARCHER DETECTION         (DIMENS*
*C      PSRANG    = RANGE OF DETECTION BY SEARCHER                               *
*C      PTE    ( ) = SIGNAL POWER SPECTRUM AS SEEN BY EVADER                     *
*C      PTS    ( ) = SIGNAL POWER SPECTRUM AS SEEN BY SEARCHER                   *
*C      PXE    ( ) =  EVADER  X-POSITION AT POINT ON BRANCH                      *
*C      PXS    ( ) = SEARCHER X-POSITION AT POINT ON BRANCH                      *
*C      PYE    ( ) =  EVADER  Y-POSITION AT POINT ON BRANCH                      *
*C      PYS    ( ) = SEARCHER Y-POSITION AT POINT ON BRANCH                      *
*C      PZE    ( ) =  EVADER  Z-POSITION AT POINT ON BRANCH                      *
*C      PZS    ( ) = SEARCHER Z-POSITION AT POINT ON BRANCH                      *
*C      Q1        = DUMMY VARIABLE                                               *
*C      Q2        = DUMMY VARIABLE                                               *
*C      RAD       =                                                              *
*C                = COMPUTATIONAL VARIABLE                                       *
*C      RANGE ( ) = HORIZONTAL RANGE BETWEEN SHIPS AT POINT ON BRANCH            *
*C      RANGEC( ) = COMPUTED RANGE FOR RAY TYPE                                  *
*C      RANGEH    = HORIZONTAL RANGE BETWEEN SHIPS                               *
*C      RANGET    = HORIZONTAL TRAVEL OF RAY PORTION                             *
*C      RC        =                                                              *
*C      RCJ       = RANGE OF CLOSE                                               *
*C      RCZ1      = CONVERGENT ZONE RANGE                                        *
*C      RCZ2      = CONVERGENT ZONE RANGE                                        *
*C      RNGMOD( , )= HORIZONTAL RANGE OF RAY FOR EACH MODE OF PROPAGATION *
*C      RNGSTR    = RANGE OF CLOSING TO INITIALIZE RNGCLS PROPERLY               *
*C      RSD       =                                                              *
```

```
*C                  COMPUTATIONAL VARIABLE                                      *
*C       RSD1       =                                                           *
*C                  COMPUTATIONAL VARIABLE                                      *
*C       SCSD       =                                                           *
*C                  COMPUTATIONAL VARIABLE                                      *
*C       SDCON      = WAVE HEIGTH PARAMETER                                     *
*C       SDCON      = SURFACE DUCT CONSTANT                                     *
*C       SDEPTH     = SEE -DEPSER-                                              *
*C       SE1        = CONSTANT SPEED OF EVADER                                (K*
*C       SIGPLS( , )= SIGNAL TO NOISE RATIO PLUS ONE FOR SHIPS           (DIMENS*
*C       SINPHI( )  = SINE OF STEERING DEPRESSION ANGLES FOR SHIPS       (DIMENS*
*C       SINRAD( )  = SINE OF AZIMUTHAL SIGNAL ARRIVAL ANGLE             (DIMENS*
*C       SLOPEJ( )  = SLOPE IN LAYER                                     (SEC**2/*
*C       SM         = SUMMATION VARIABLE                                        *
*C       SMPEDT     = SUMMATION FOR COMPUTING AVERAGE EVADED DETECTI(DIMENS*
*C       SMPSDT     = SUMMATION FOR COMPUTING AVERAGE SEARCHER DETEC(DIMENS*
*C       SMTCNE     = TOTAL TIME EVADER IS IN CONTACT WITH SEARCHER             *
*C       SMTCNS     = TOTAL TIME SEARCHER IS IN CONTACT WITH EVADER             *
*C       SPAFUN     = FUNCTION TO COMPUTE  THE SPA REQUIRED                     *
*C       SPATEV     = PROPAGATION SPEED AT EVADER                             (K*
*C       SPATSE     = PROPAGATION SPEED AT SEARCHER                             *
*C       SPDTER( )  = SPEED OF PROPAGATION AT SHIPS                           (K*
*C       SPDVER     = PROPAGATION SPEED AT VERTEX OF RAY                      (K*
*C       SPI( )     = SPREADING LOSS CONSTANT                            (DIMENS*
*C       SPT( )     = SPREADING LOSS CONSTANT                            (DIMENS*
*C       SPVRSQ     = VERTEX VELOCITY OF RAY SQUARED                     ((K-YD2*
*C       SQAREA( )  = ARRAY CONSTANT                                     (YD-S*
*C       SQRTZL     -                                                           *
*C                  COMPUTATIONAL VARIABLE                                      *
*C       SS1        = CONSTANT SPEED OF SEARCHER                              (K*
*C       STATD      = SEE -NDSTAT-                                              *
*C       STATE      = SEE -NESTAT-                                              *
*C       STNPEV     = SIGNAL TO NOISE RATIO PLUS ONE FOR EVADER          (DIMENS*
*C       STNPSE     = SIGNAL TO NOISE RATIO PLUS ONE FOR SEARCHER        (DIMENS*
*C       SUMEVA     = ACCUMLATIVE PROBABILITY OF EVASION                 (DIMENS*
*C       SUMKIL     = ACCUMLATIVE PROBABILITY OF EVASION                 (DIMENS*
*C       SUMPTH     = SUMMATION OF PATH PROBABILITIES AT BRANCH ENDS(DIMENS*
*C       SUMTOT     = A SUMMATION OF ALL POSSIBILE PROBABILITIES         (DIMENS*
*C       SURANG( )  = SURFACE ANGLE FOR THE RAY TYPE                          (*
*C                  NOT USED OR REQUIRED                                        *
*C       TARRIV( , )= TRANSFORMED ARRIVAL ANGLE VECTOR                   (DIMENS*
*C       TCONEN( )  = TIME  EVADER  IS IN CONTACT AT BRANCH POINT               *
*C       TCONSN( )  = TIME SEARCHER IS IN CONTACT AT BRANCH POINT               *
*C       TCZAV1     =             AVERAGE OF CONVERGENT ZONE ANGLES           (*
*C       TCZAV2     =             AVERAGE OF CONVERGENT ZONE ANGLES           (*
*C       TI         = STARTING ANGLE (IN DEGREES) OF RAY FROM SEARCHER          *
*C       TI         = TIME INCREMENT FOR RAY TO GO X K-YD                       *
*C       TIMCON     = PROPAGATION TIME OF RAY                                   *
*C       TIR   ( )  = INITIAL ANGLE OF RAY WITH RANGEH                        (*
*C       TMATEV( , )= TRANSFORMATION MATRIX FOR EVADER                   (DIMENS*
*C       TMATRX(,,) = TRANSFORMATION MATRIX                              (DIMENS*
*C       TMATSE( , )= TRANSFORMATION MATRIX FOR EVADER                   (DIMENS*
*C       TSTOKY     = FUNCTION TO CHANGE KNOTS TO KILO-YARDS/SECOND             *
*C       TTR   ( )  = ARRIVAL ANGLE OF RAY TYPE                               (*
*C       TVECTR( , )= TRANSFORMED BEAM-STEERING-DIRECTION VECTOR         (DIMENS*
*C       TWOALE     = TWO TIMES ALPHAE**2                                (DIMENS*
*C       TWOALS     = TWO TIMES ALPHAS**2                                (DIMENS*
*C       UC         = VELOCITY OF PROPAGATION (IN K-YD/SEC) AT POINT IN LAY*
*C       V          = PROPAGATION SPEED AT END POINT OF RAY                   (K*
*C       VAREVA( )  = VARIANCE OF SMOOTHED S/N FOR EVADER                       *
*C       VARSER( )  = VARIANCE OF SMOOTHED S/N FOR SEARCHER                     *
*C       VXE   ( )  = X-COMPONENT OF EVADER VELOCITY AT POINT ON BRANCH  (K*
*C       VXEINT     = INITIAL EVADER SPEED IN X-DIRECTION                     (K*
*C       VXS   ( )  = X-COMPONENT OF SEARCHER VELOCITY AT POINT ON BRANCH(K*
*C       VYE   ( )  = Y-COMPONENT OF EVADER VELOCITY AT POINT ON BRANCH  (K*
*C       VYEINT     = INITIAL EVADER SPEED IN Y-DIRECTION                     (K*
*C       VYS   ( )  = Y-COMPONENT OF SEARCHER VELOCITY AT POINT ON BRANCH(K*
*C       X          = RANGE (IN K-YD) OF HORIZONTAL TRAVEL OF RAY WITHIN LA*
*C       X          = FRACTIONAL PART OF LAYER                                  *
*C       XE         = SIGNAL TO NOISE RATIO AT EVADER                    (DIMENS*
*C       XS         = SIGNAL TO NOISE RATIO AT SEARCHER                  (DIMENS*
*C      Y1            =                                                         *
*C      Y2            =                                                         *
*C      Z1            = STARTING DEPTH (IN K-YD) OF RAY                         *
*C      Z1B           = STARTING DEPTH (IN K-YD) OF CONVERGENT ZONE RAY         *
*C      Z1B           = DEPTH (IN K-YD) OF EVADER                               *
*C      Z2            = ENDING DEPTH (IN K-YD) OF RAY                           *
*C      Z2B           = ENDING DEPTH (IN K-YD) OF CONVERGENT ZONE RAY           *
*C      Z2B           = DEPTH OF EVADER                                         *
*C      ZB            = DEPTH (IN K-YD) OF POINT WITHIN LAYER TO WHICH RAY IS*
*C      ZL            = DEPTH OF MAX OR MIN POINT IN VELOCITY PROFILE           *
*C      ZONACZ        = MODIFIED VALUE OF ACZ FOR SIMPLIFICATION      (DIMENS*
*C      ZONBCZ        = MODIFIED VALUE OF BCZ FOR SIMPLIFICATION      (DIMENS*
*C      ZSTART        = STARTING DEPTH OF RAY PORTION                           *
*C      ZVLO          = DEPTH (IN K-YD) OF VERTEX POINT OF CONVERGENT RAY       *
*C      ZVUP          = DEPTH OF UPPER VERTEX POINT FOR RAY                     *
*C  *                                                                           *
*C  ****************************************************************************
*C                                                                              *
*C                                                                              *
*       NAMELIST                                                                *
*      N / AVERAG /                                                             *
*      N SUMTOT, SUMKIL, SUMEVA, SUMPTH,                                        *
*      N TRENOR, TREKIL, TREEVA, TREPTH,                                        *
*      N CONNOR, AVEKIL, AVEEVA, AVEPTH,                                        *
*      N SMPEDT, SMPSDT, SMTCNS, SMTCNE, PERANG, PSRANG                         *
*      N / BOTTOM /                                                             *
*      N TIR, TTR, RANGEC, PATHLN, DRDXDC, BOTANG, SPI, SPT, RANGEH             *
*      N / BRANCH /                                                             *
*      N NEI, NSI, NR, SUMKIL, BRNKIL, SUMEVA, BRNEVA, AVBREV, AVBRKL           *
*      N / CHECKS /                                                             *
*      N RCJ, HE1                                                               *
*      N / CZONEN /                                                             *
*      N AMLSRD, HCI, HZSD, SDCON, NCONSD,                                      *
*      N RCZ1, RCZ2, TCZAV1, TCZAV2                                             *
*       NAMELIST                                                                *
*      N / DATAIN /                                                             *
*      N ACZ   , ADI1  , ADI2  , ALPHAE, ALPHAS, ANGMAX, AR1X  , AR1Y  ,        *
*      N AR1Z  , AR2X  , AR2Y  , AR2Z  , ARRAY1, ARRAY2, ARRYH1, ARRYH2,        *
*      N ARRYW1, ARRYW2, B2E   , B2S   , BCZ   , BETAE , BETAS , BF1   ,        *
*      N BF2   , CONLR1, CONLR2, DELANG, DELF  , DELHED, DELRNG, DELTAE,        *
*      N DELTAS, DEPEVA, DEPKYD, DEPSER, DI1   , DI2   , DIFTI , DTRAD ,        *
*      N DPFTEV, DPFTSE,                                                        *
*      N E11   , E12   , E13   , E21   , E22   , E23   , FOE   , FOS   ,        *
*      N F1E   , F1S   , F2E   , F2S   , FBWE  , FBWS  , FLN1  , FLN2  ,        *
*      N FNE   , FNS   , FRES1 , FRES2 , FXE   , FXS   , HDINEV, HEDMAX,        *
*      N HS1   , IO1   , IO2   , MAXLAY, MAXTIM, MECO  , NEMAX , NPCO  ,        *
*      N NEWLAY,                                                                *
*      N NPRINT,                                                                *
*      N NSMAX , NULAPO, PDEMIN, PDSMIN, PHIE  , PHIS  , POR   , PPAMIN,        *
*      N PRE   , PRK   , QTRAN1, QTRAN2, RGINEV, RI    , RNGMAX, SPDEVA,        *
*      N SS    ,                                                                *
*      N SPDKYD, SPDSER, THREVA, THRSER, WRANGE, ZW                             *
*       NAMELIST                                                                *
*      N / DATAOU /                                                             *
*      N ADI1, ADI2, DI1, DI2, IO1, IO2,                                        *
*      N BF1, BF2, CONLR1, CONLR2, DEPKYD, FLN1, FLN2, FNE, FNS, FXE, FXS,*
*      N ARRAYD,                                                                *
*      N FULANG,                                                                *
*      N SPDKYD, DUMMY1                                                         *
*      N / FVALUE /                                                             *
*      N COSAVE, DELRCZ, DYHITH, ARAREA, CTWOPI, SQAREA, ANGDEP, ARWIDT,        *
*      N FRQLCN, FRQIDR, FRQTRN, FRQASD, FRQNCN, NUMFRQ                         *
*       NAMELIST                                                                *
*      N / INPUTS /                                                             *
*      N ACZ, BCZ,                                                              *
*      N ALPHAE, ALPHAS, ANGMAX, ARRAY1, ARRAY2, ARRYH1, ARRYH2, ARRYW1,        *
*      N AR1X, AR1Y, AR1Z, AR2X, AR2Y, AR2Z,                                    *
*      N ARRYW2, B2E   , B2S   , BETAE , BETAS , DELANG, DELF  , DELHED,        *
*      N DPFTEV, DPFTSE,                                                        *
*      N DELRNG, DELTAE, DELTAS, DEPEVA, DEPSER, DIFTI , DTRAD , FOE   ,        *
*      N FOS   , F1E   , F1S   , F2E   , F2S   , FBWE  , FBWS  , FRES1 ,        *
*      N FRES2 , HDINEV, HS1   , MAXLAY, MAXTIM, MECO  , NEMAX , NPCO  ,        *
*     N NPRINT,                                                              *
*     N NEWLAY,                                                              *
*     N NSMAX , NULAPO, PHIE  , PHIS  , POR   , PRE   , PRK   , QTRAN1,      *
*     N PPAMIN, PDSMIN, PDEMIN,                                              *
*     N QTRAN2, RGINEV, RI    , RNGMAX, SPDEVA, SPDSER, WRANGE, ZW           *
*     N,THREVA, THRSER                                                       *
*      NAMELIST                                                              *
*     N / NOISES /                                                           *
*     N XE, XS                                                               *
*     N / POWERS /                                                           *
*     N PTE, PTS, PNE, PNS                                                   *
*     N / TABLES /                                                           *
*     N PDS, PGS, PGE                                                        *
*C                                                                           *
*      TSTOKY( DUMMY1 ) = DUMMY1*5.6266667E-4                                *
*      FTTOKY( DUMMY1 ) = DUMMY1/3000.0                                      *
*C                                                                           *
*      NEWLAY = -1                                                           *
*      DGPRRD = 57.2957795                                                   *
*      CONSPI = 3.14159265                                                   *
*      CTWOPI = 2.0*CONSPI                                                   *
*C                                                                           *
******************************************************************************
                                      I
                                      0(............................................0
                                      I                                             I
******************************************************************************      I
*   10 CONTINUE                                                              *      I
*      READ( 5, DATAIN )                                                     *      I
*      WRITE ( 6, 1000 )                                                     *      I
*      WRITE( 6, INPUTS )                                                    *      I
*C                                                                           *      I
*C     SET UP LAYER CONSTANSIS AND DIFFERENT TYPES OF RAYS                   *      I
*C                                                                           *      I
*      DELRAD = DELANG/DGPRRD                                                *      I
*      SE1 = TSTOKY( SPDEVA )                                                *      I
*      SS1 = TSTOKY( SPDSER )                                                *      I
******************************************************************************      I
                                      I                                             I
                                      I                                             I
                                      I                                             I
                                      I                                             I
                                      I                                             I
******************************************************************************      I
*      IF( NPCO .NE. 2 )                                                     *......I.......0
*     * GO TO 270                                                            *      I       I
******************************************************************************      I       I
                                      I                                             I       I
                                      I                                             I       I
                                      I                                             I       I
******************************************************************************      I       I
*      IF( MECO .NE. 2 )                                                     *......I.......I.......0
*     * GO TO 260                                                            *      I       I       I
******************************************************************************      I       I       I
                                      I                                             I       I       I
                                      I                                             I       I       I
                                      I                                             I       I       I
******************************************************************************      I       I       I
*      WRITE ( 6, 10000 )                                                    *      I       I       I
******************************************************************************      I       I       I
                                      I                                             I       I       I
                                      I                                             I       I       I
                                      I                                             I       I       I
******************************************************************************      I       I       I
*       GO TO 10                                                             *.....)A       I       I
******************************************************************************      I       I       I
                                                                                    I       I       I
```

```
                                                                                   I       I        I
*****************************************************************************      I       I        I
*C                                                                          *      I       I        I
*****************************************************************************      I       I        I
                                      I                                            I       I        I
                                      0(...........................................I.......I........0
                                      I                                            I       I
*****************************************************************************      I       I
*  260 CONTINUE                                                             *      I       I
*****************************************************************************      I       I
                                      I                                            I       I
                                      I                                            I       I
                                      I                                            I       I
*****************************************************************************      I       I
*       IF( SS1 .GT. SH1 )                                                  *......I......)V
*       * GO TO 270                                                         *      I       I
*****************************************************************************      I       I
                                      I                                            I       I
                                      I                                            I       I
*****************************************************************************      I       I
*       WRITE ( 6,11000 )                                                   *      I       I
*****************************************************************************      I       I
                                      I                                            I       I
                                      I                                            I       I
                                      I                                            I       I
*****************************************************************************      I       I
*       GO TO 10                                                            *.....)A       I
*****************************************************************************      I       I
                                                                                   I       I
*****************************************************************************      I       I
*C                                                                          *      I       I
*****************************************************************************      I       I
                                      I                                            I       I
                                      0(...........................................I.......0
                                      I                                            I
*****************************************************************************      I
*  270 CONTINUE                                                             *      I
*****************************************************************************      I
                                      I                                            I
                                      I                                            I
*****************************************************************************      I
*       IF( NEWLAY )                                                        *......I.......0
*       * 290, 280, 380                                                     *......I.......I........0
*****************************************************************************      I       I        I
                                      I                                            I       I        I
                                      I                                            I       I        I
*****************************************************************************      I       I        I
*  290 CONTINUE                                                             *      I       I        I
*       CALL                                                                *      I       I        I
*      S             LAYERS                                                 *      I       I        I
*      S             ( MAXLAY )                                             *      I       I        I
*****************************************************************************      I       I        I
                                      I                                            I       I        I
                                      I                                            I       I        I
*****************************************************************************      I       I        I
*       IF( NULAPO .GE. MAXLAY )                                            *......I.......I........I.......0
*       * GO TO 20                                                          *      I       I        I       I
*****************************************************************************      I       I        I       I
                                      I                                            I       I        I       I
                                      0(...........................................I.......0        I       I
                                        I                              I              I       I
          *************************************************************    I              I       I
          *  280 CONTINUE                                             *    I              I       I
          *      DEPEVA = FTTOKY( DPFTEV )                            *    I              I       I
          *      DEPSER = FTTOKY( DPFTSE )                            *    I              I       I
          *      CALL                                                 *    I              I       I
          *     S            RAYCTL                                   *    I              I       I
          *      CALL                                                 *    I              I       I
          *     S            MSTRAY                                   *    I              I       I
          *************************************************************    I              I       I
                                        I                              I              I       I
                                        I                              I              I       I
                                        I                              I              I       I
                                        I                              I              I       I
          *************************************************************    I              I       I
          *      IF( NPRINT .LT. 1 )                                  *....I.............)V       I
          *     * GO TO 380                                           *    I              I       I
          *************************************************************    I              I       I
                                        I                              I              I       I
                                        I                              I              I       I
          *************************************************************    I              I       I
          *      WRITE( 6, CZONEN )                                   *    I              I       I
          *************************************************************    I              I       I
                                        I                              I              I       I
                                        I                              I              I       I
          *************************************************************    I              I       I
          *      IF( NPRINT .LT. 3 )                                  *....I.............)V       I
          *     * GO TO 380                                           *    I              I       I
          *************************************************************    I              I       I
                                        I                              I              I       I
                                        0(................................................I.......0
                                        I                              I              I
          *************************************************************    I              I
          *   20 CONTINUE                                             *    I              I
          *      NUMLAY = NULAPO - 1                                  *    I              I
          *       WRITE ( 6, 2000 )                                   *    I              I
          *************************************************************    I              I
                                        I                              I              I
                                        I                              I              I
          *************************************************************    I              I
I.........*      DO 100                                               *    I              I
I         *     I            I = 1,NUMLAY                             *    I              I
I         *************************************************************    I              I
I                                       I                              I              I
I                                       I                              I              I
I         *************************************************************    I              I
I         *      VELMID = FVELOC( DELTAZ(I)/2.0, I )                  *    I              I
I         *      WRITE ( 6, 7000 )                                    *    I              I
I         *     W I,                                                  *    I              I
I         *     W DEPKYD(I),                                          *    I              I
I         *     W DELTAZ(I),                                          *    I              I
I         *     W SPDKYD(I),                                          *    I              I
I         *     W CONSV0(I),                                          *    I              I
I         *     W CONSG0(I),                                          *    I              I
I         *     W CONSG1(I),                                          *    I              I
I         *     W CONSG2(I),                                          *    I              I
I         *     W SLOPEJ(I),                                          *    I              I
I         *     W VELMID                                              *    I              I
I         *      X = DELTAZ(I)/20.0                                   *    I              I
I         *      VELOCI = SPDKYD(I)                                   *    I              I
I         *      D = 0.0                                              *    I              I
I         *************************************************************    I              I
I                                       I                              I              I
```

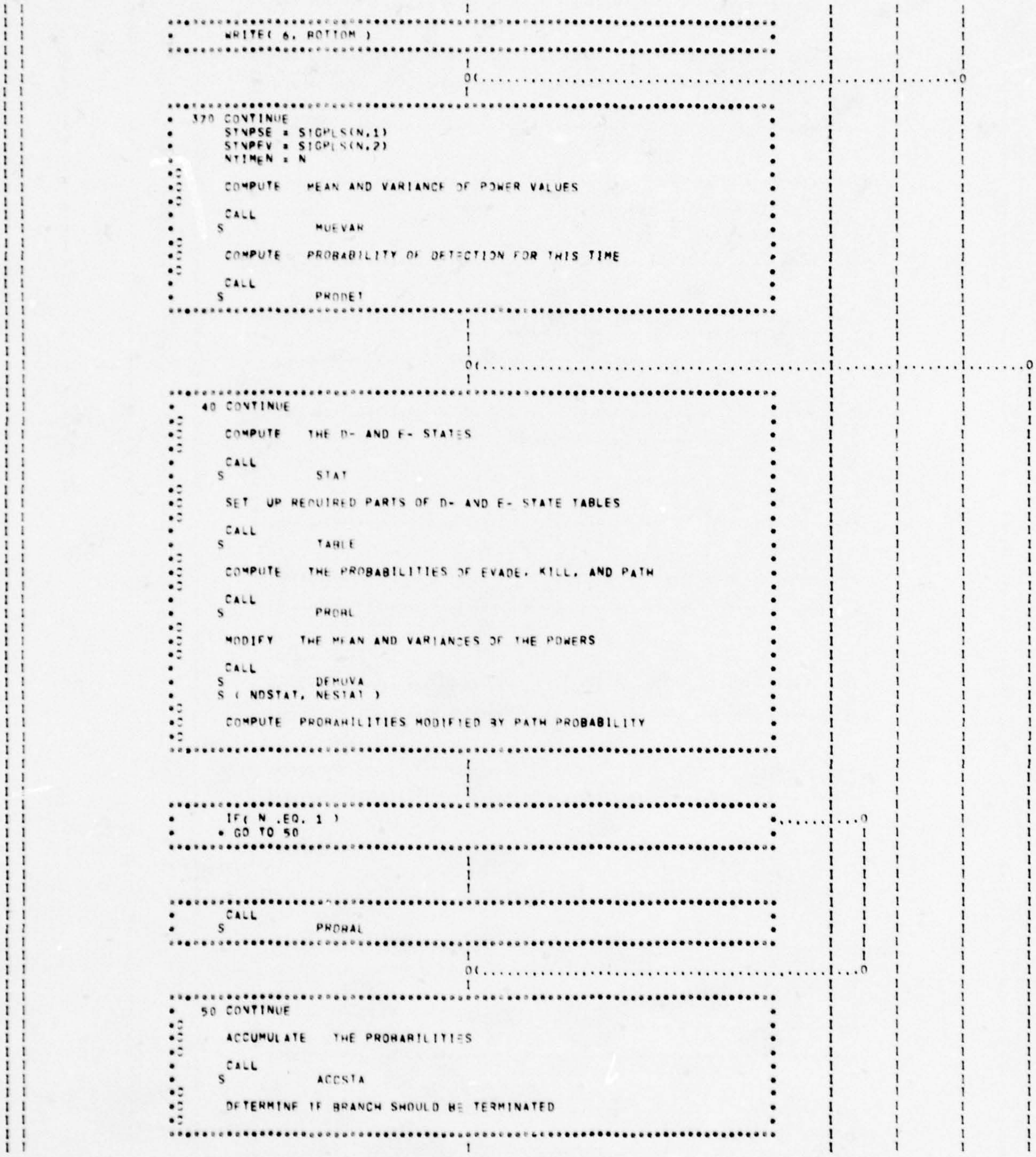
```
      DO 100
      I             J = 1,20

      D = D + X
      UC = FVELOC( D, I )
      DELTAV = VELOCI - UC
      VELOCI = UC
      SLOPEN = DELTAV/X
      WRITE ( 6, 3000 )
     W UC
     W , DELTAV, SLOPEN

  100 CONTINUE

      WRITE( 6, 7000 )
     W NULAPO,
     W DEPKYD(NULAPO),
     W DELTAZ(NULAPO),
     W SPDKYD(NULAPO),
     W CONSVO(NULAPO),
     W CONSGO(NULAPO)
      IF( NULAPO .GE. MAXLAY )  CALL  DUMP
      N1 = NUANMO + 1
      WRITE( 6, 4000 )
      WRITE ( 6, 5000 )
     W ( ANGINT(J), ( RNGMOD(I,J), I = 1,6 ), J = 1,N1 )

  380 CONTINUE
C
C     SET UP CONSTANTS USED IN PROGRAM
C
      NEMAX = MAXTIM
      NSMAX = MAXTIM
      ZONACZ = 60.0 - ACZ
      ZONBCZ = BCZ - ACZ
      AMAXNN = MAXTIM
      BLA1 = POR/0.24
      BLA2 = ( 1.0 - BLA1 )/3.125/CONSPI/CONSPI
      BLA3 = 6.0 + 22.0*( POR - 0.27 )
      DELRCZ = RCZ2 - RCZ1
      EDEPTH = DEPEVA
      HCI = 30.0 + HCI + HZSD
      RNGSTR = RGINEV - DELRNG
      SDEPTH = DEPSER
      WAVECN = 1.035*TNLG10( ZW ) - 44.0
      ONMIAE = 1.0 - ALPHAE
      A1 = ONMIAE*ONMIAE
      FALPHE = A1/( 1.0 - A1 )
      TWOALE = 2.0*ALPHAE*ALPHAE
      THREVA = AMIN1( THREVA, 1.4142*TWOALE*FALPHE/A1 + 1.0 )
      ONMIAS = 1.0 - ALPHAS
      A1 = ONMIAS*ONMIAS
      FALPHS = A1/( 1.0 - A1 )
      TWOALS = 2.0*ALPHAS*ALPHAS
      THRSER = AMIN1( THRSER, 1.4142*TWOALS*FALPHS/A1 + 1.0 )
      ANGDEP(1) = PHIS
      ANGDEP(2) = PHIE
      FRQLOW(1) = F1S
      FRQLOW(2) = F1E
      NARRAY(1) = ARRAY1
      NARRAY(2) = ARRAY2
      NUMFRQ(1) = ( F2S - F1S )/DELF + 1.0
      NUMFRQ(2) = ( F2E - F1E )/DELF + 1.0
      SPDTER(1) = SPATSE
      SPDTER(2) = SPATEV
```

```
                                                       I                                               I
        ***********************************************************************************            I
I.........*        DO 400                                                                  *           I
I         *          I            M = 1,2                                                  *           I
I         ***********************************************************************************          I
I                                                      I                                               I
I                                                      I                                               I
I                                                      I                                               I
I         ***********************************************************************************          I
I         *        COSAVE(M) = COS( ANCZAV(M) )                                             *          I
I         *        COSPHI(M) = COS( ANGDEP(M) )                                             *          I
I         *        SINPHI(M) = SIN( ANGDEP(M) )                                             *          I
I         *C                                                                                *          I
I         *C       SET UP TRANSFORMATION MATRICES                                           *          I
I         *C                                                                                *          I
I         *        MS = M + 2                                                               *          I
I         ***********************************************************************************          I
I                                                      I                                               I
I                                                      I                                               I
I                                                      I                                               I
I         ***********************************************************************************          I
I I.......*        DO 200                                                                   *          I
I I       *          I            I = 1,3                                                   *          I
I I       ***********************************************************************************          I
I I                                                    I                                               I
I I                                                    I                                               I
I I                                                    I                                               I
I I       ***********************************************************************************          I
I I       *        ARRAYD(I,M) = CONSPI*( AINPUT(I,M)/3.0 )/SPDTER(M)                       *          I
I I       *        A1 = EANGLE(I,M)/DGPRRD                                                  *          I
I I       *        EULANG(I,MS) = SIN( A1 )                                                 *          I
I I       *        EULANG(I,M)  = COS( A1 )                                                 *          I
I I       ***********************************************************************************          I
I I                                                    I                                               I
I I                                                    I                                               I
I I                                                    I                                               I
I I                                                    I                                               I
I I       ***********************************************************************************          I
I  .......*  200 CONTINUE                                                                   *          I
I         ***********************************************************************************          I
I                                                      I                                               I
I                                                      I                                               I
I                                                      I                                               I
I         ***********************************************************************************          I
I         *        A1 = EULANG(3,M)*EULANG(1,M)                                             *          I
I         *        A2 = EULANG(3,MS)*FULANG(1,MS)                                           *          I
I         *        A3 = EULANG(3,M)*EULANG(1,MS)                                            *          I
I         *        A4 = EULANG(3,MS)*EULANG(1,M)                                            *          I
I         *        TMATRX(1,1,M) = A1 - A2*FULANG(2,M)                                      *          I
I         *        TMATRX(1,2,M) = A3 + A4*EULANG(2,M)                                      *          I
I         *        TMATRX(1,3,M) = FULANG(2,MS)*EULANG(3,MS)                                *          I
I         *        TMATRX(2,1,M) =-A3*EULANG(2,M) - A4                                      *          I
I         *        TMATRX(2,2,M) = A1*EULANG(2,M) - A2                                      *          I
I         *        TMATRX(2,3,M) = FULANG(2,MS)*EULANG(3,M)                                 *          I
I         *        TMATRX(3,1,M) =  EULANG(2,MS)*EULANG(1,MS)                               *          I
I         *        TMATRX(3,2,M) = -EULANG(2,MS)*EULANG(1,M)                                *          I
I         *        TMATRX(3,3,M) = FULANG(2,M)                                              *          I
I         *        ANDISO(1,M) = DIRANG(1,M)/DGPRRD                                         *          I
I         *        N2 = NUDIAN(M)                                                           *          I
I         ***********************************************************************************          I
I                                                      I                                               I
I                                                      I                                               I
I                                                      I                                               I
I         ***********************************************************************************          I
I I.......*        DO 300                                                                   *          I
I I       *        I            I1 = 2,N2                                                   *          I
I I       ***********************************************************************************          I
I I                                                    I                                               I
I I                                                    I                                               I
I I                                                    I                                               I
I I       ***********************************************************************************          I
I I       *        I = I1 - 1                                                               *          I
I I       *        ANDISO(I1,M) = DIRANG(I1,M)/DGPRRD                                       *          I
I I       *        DLDIAN(I,M) = ANDISO(I1,M) - ANDISO(I,M)                                 *          I
I I       ***********************************************************************************          I
I I                                                    I                                               I
I I                                                    I                                               I
```

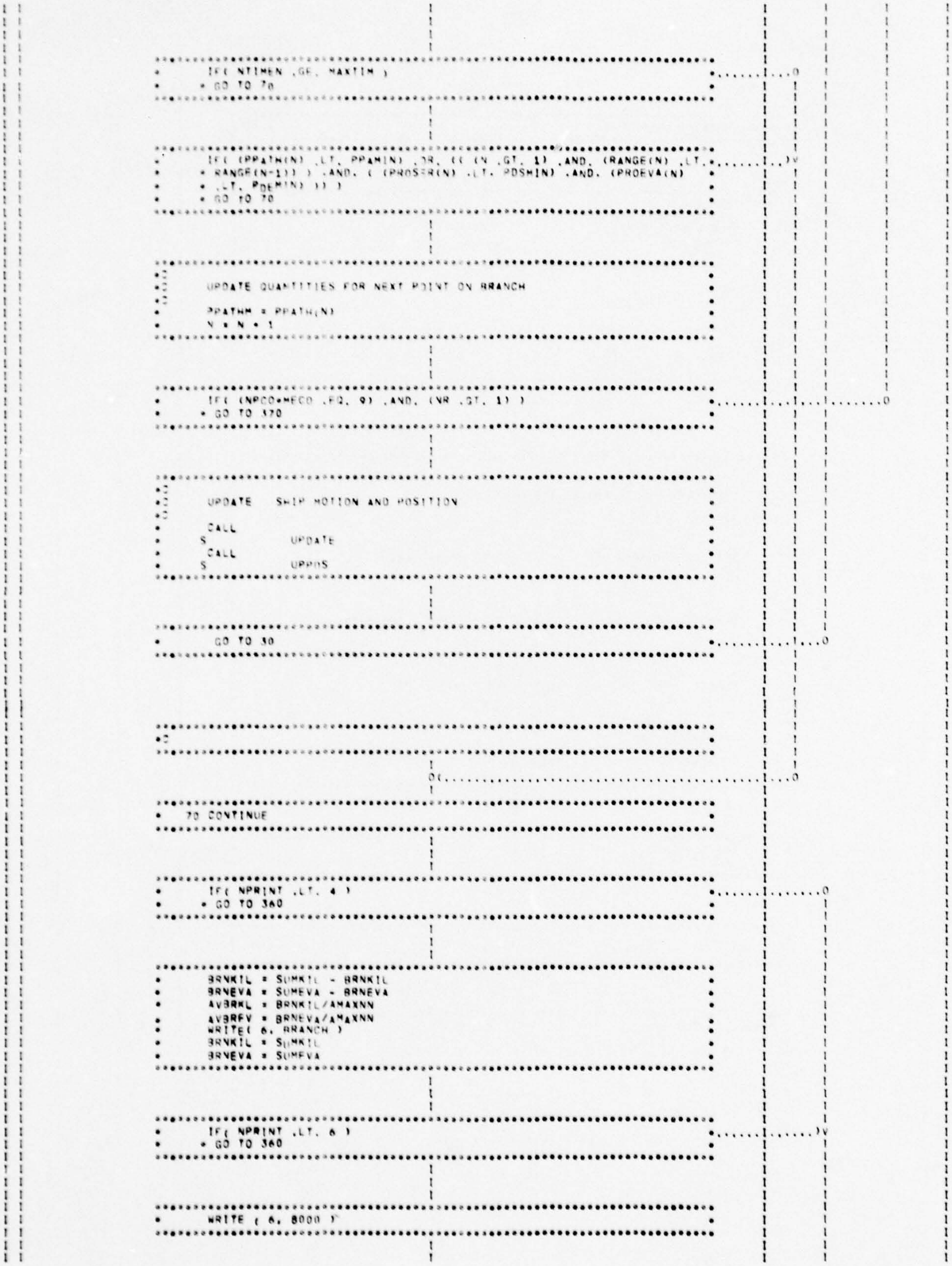
```
I I                                     I                                           I
I I        *************************************************************************        I
I ........* 300 CONTINUE                                                          *        I
I          *************************************************************************        I
I                                       I                                           I
I                                       I                                           I
I                                       I                                           I
I          *************************************************************************        I
I          *C                                                                     *        I
I          *C     SET UP THE DIFFERENT FREQUENCY CONSTANTS                        *        I
I          *C                                                                     *        I
I          *      F = FRQLOW(M)                                                   *        I
I          *      FRQLOW(M) = FRQLOW(M) - DELF                                    *        I
I          *      N1 = NUMFRQ(M) + 1                                              *        I
I          *************************************************************************        I
I                                       I                                           I
I                                       I                                           I
I                                       I                                           I
I                                       I                                           I
I                                       I                                           I
I          *************************************************************************        I
I I.......*      DO 400                                                          *        I
I I        *    I           J = 1,N1                                              *        I
I I        *************************************************************************        I
I I                                     I                                           I
I I                                     I                                           I
I I                                     I                                           I
I I        *************************************************************************        I
I I        *      FRQSIG(J,M) = F                                                 *        I
I I        *      FRQNOS(J,M) = F                                                 *        I
I I        *      A2 = SQRT(F)                                                    *        I
I I        *      FRQASD(J,M) = SDCON*A2                                          *        I
I I        *      A1 = F*F                                                        *        I
I I        *      FRQLCN(J,M) = ( 40.0/( 4100.0 + A1 ) + 2.75E-4 )*A1             *        I
I I        *      A1 = F/FRQRES(M)                                                *        I
I I        *      A2 = 1.0/(1.0 +(ARQTRN(M)*( A1 - 1.0/A1 ) ) **2 )               *        I
I I        *      FRQTRN(J,M) = A2                                                *        I
I I        *      A3 = TNLG10( F )                                                *        I
I I        *      FRQNCN(J,M)=(ALOGIN(WAVECN-1.667*A3)/2.0+ALOGIN(FLONOS(J,M)))*A2 *        I
I I        *      FRQIDB(J,M) = A3 + BLA3                                         *        I
I I        *      DELBAF(J,M) = HAFFUN(J+1,M) - BAFFUN(J,M)                       *        I
I I        *      F = F + DELF                                                    *        I
I I        *************************************************************************        I
I I                                     I                                           I
I I                                     I                                           I
I I                                     I                                           I
I I        *************************************************************************        I
..........* 400 CONTINUE                                                          *        I
           *************************************************************************        I
                                        I                                           I
                                        I                                           I
                                        I                                           I
           *************************************************************************        I
           *      ARRAYD(1,1) = ARRAYD(1,1)/ARRAY1                                *        I
           *      ARRAYD(1,2) = ARRAYD(1,2)/ARRAY2                                *        I
           *      N1 = MAX0( NUMFRQ(1), NUMFRQ(2) ) + 2                           *        I
           *************************************************************************        I
                                        I                                           I
                                        I                                           I
                                        I                                           I
           *************************************************************************        I
I.........*      DO 500                                                          *        I
I          *    I           J = 1,N1                                              *        I
I          *************************************************************************        I
I                                       I                                           I
I                                       I                                           I
I                                       I                                           I
I          *************************************************************************        I
I          *      FNE(J) = FRQNOS(J,2)                                            *        I
I          *      FNS(J) = FRQNOS(J,1)                                            *        I
I          *      FXE(J) = FRQSIG(J,2)                                            *        I
I          *      FXS(J) = FRQSIG(J,1)                                            *        I
I          *************************************************************************        I
I                                       I                                           I
```

```
  500 CONTINUE

C
C     SET UP CONSTANTS USED IN D- AND E- STATE TABLES
C
      PF(1)=PRF
      PGE(1)=1.
      PGS(1)=1.
      PKDS(1)=PRK
C
      PDE(2)=1.
      PE(2)=PRF
      PGS(2)=1.
      PKDS(2)=0.
      PRNOEV(2) = 0.0
      PRYSEV(2) = 1.0
C
      PDE(3)=0.
      PDS(3)=1.
      PF(3)=0.
      PKDS(3)=PRK
      PRNOEV(3) = 1.0
      PRNOSE(3) = 0.0
      PRYSEV(3) = 0.0
      PRYSSE(3) = 1.0
C
      PDS(4)=1.
      PKDS(4)=0.
      PRNOSE(4) = 0.0
      PRYSSE(4) = 1.0
C
      PDS(5)=0.
      PKDS(5)=0.
      PRNOSE(5) = 1.0
      PRYSSE(5) = 0.0
C
C     INITIALIZE SUMMATION CONSTANTS TO ZERO
C
      BRNEVA = 0.0
      BRNKIL = 0.0
      PERANG = 0.0
      PSRANG = 0.0
      SMPEDT = 0.0
      SMPSDT = 0.0
      SMRANG = 0.0
      SMTCNE = 0.0
      SMTCNS = 0.0
      SUMEVA = 0.0
      SUMPTH = 0.0
      SUMKIL = 0.0
C
C     START LOOPS THROUGH HEADINGS AND RANGES
C
      HF1 = HDINFV
      NUMHED = ( HEDMAX - HDINEV )/DELHED + 1.0
      NUMRNG = ( RNGMAX - RGINEV )/DELRNG + 1.0
      FRLOEV = F1E - DELF
      FRLOSE = F1S - DELF

      DO 110
        I             K1= 1,NUMHED
```

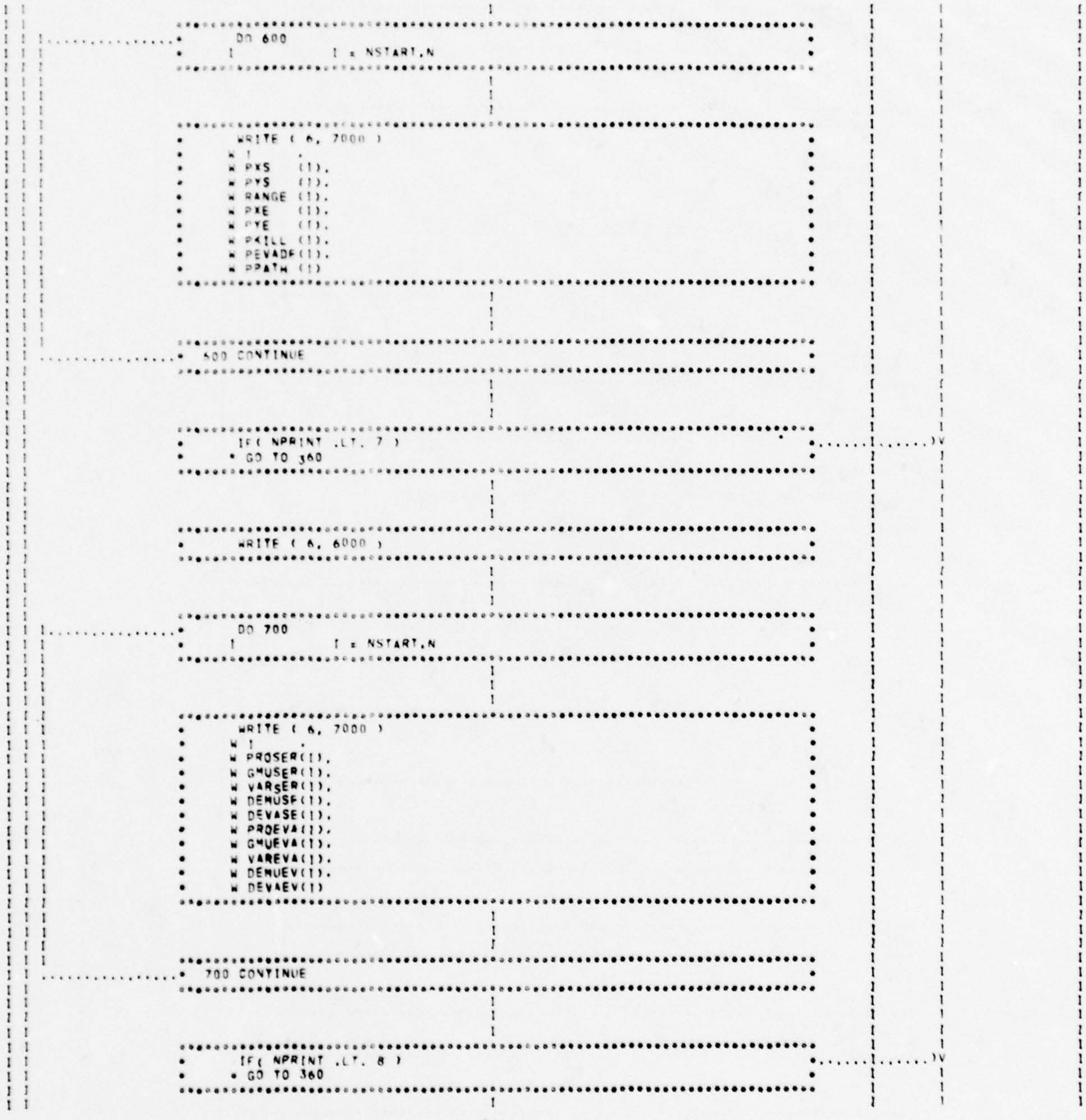
```
I                                                          I                                             I
I                    ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                    •      RCJ = RNGSTR                                                          •      I
I                    ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                                                          I                                             I
I                                                          I                                             I
I                                                          I                                             I
I                    ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I I..................•      DO 900                                                                •      I
I I                  •     I              J1= 1,NUMRNG                                            •      I
I I                  ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I I                                                        I                                             I
I I                                                        I                                             I
I I                                                        I                                             I
I I                  ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I I                  •      RCJ = RCJ + DELRNG                                                    •      I
I I                  •       WRITE ( 6, 1000 )                                                    •      I
I I                  •      WRITE( 6, CHECKS )                                                    •      I
I I                  •      NR = 1                                                                •      I
I I                  •      NEI = 1                                                               •      I
I I                  •      NSI = 1                                                               •      I
I I                  •      N = 1                                                                 •      I
I I                  •      NSTART = 1                                                            •      I
I I                  •      PPATHM = 1.0                                                          •      I
I I                  •C                                                                           •      I
I I                  •C     INITIALIZE   SEARCH PHASE AND POSITIONS                               •      I
I I                  •C                                                                           •      I
I I                  •      CALL                                                                  •      I
I I                  •     S            INIT                                                      •      I
I I                  ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I I                                                        I                                             I
I I                                                        O(............................................I.......O
I I                                                        I                                             I       I
I I                  ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I
I I                  •    30 CONTINUE                                                             •      I       I
I I                  •C                                                                           •      I       I
I I                  •C     COMPUTE   HEARING ANGLES (IN RADIANS)                                 •      I       I
I I                  •C                                                                           •      I       I
I I                  •      CALL                                                                  •      I       I
I I                  •     S            RELHR                                                     •      I       I
I I                  •      RANGEH = RANGE(N)                                                     •      I       I
I I                  •      CALL                                                                  •      I       I
I I                  •     S            STEFRA                                                    •      I       I
I I                  •C                                                                           •      I       I
I I                  •C     COMPUTE  RAY TYPES AVAILABLE                                          •      I       I
I I                  •C                                                                           •      I       I
I I                  •      CALL                                                                  •      I       I
I I                  •     S            RAYNOW                                                    •      I       I
I I                  •C                                                                           •      I       I
I I                  •C     COMPUTE   RECIEVED SIGNAL AND NOISE                                   •      I       I
I I                  •C                                                                           •      I       I
I I                  •      CALL                                                                  •      I       I
I I                  •     S            RECIEV                                                    •      I       I
I I                  •C                                                                           •      I       I
I I                  •C     COMPUTE   SIGNAL TO NOISE RATIO OF RECIEVED POWER FOR BOTH SHIPS      •      I       I
I I                  •C                                                                           •      I       I
I I                  •      CALL                                                                  •      I       I
I I                  •     S            PSIGP                                                     •      I       I
I I                  •C                                                                           •      I       I
I I                  •C     COMPUTE   SIGNAL PLUS NOISE OVER NOISE RATIO                          •      I       I
I I                  •C                                                                           •      I       I
I I                  •      SIGPLS(N,1) =                                                         •      I       I
I I                  •     E XS                                                                   •      I       I
I I                  •      SIGPLS(N,2) =                                                         •      I       I
I I                  •     E XE                                                                   •      I       I
I I                  ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I
I I                                                        I                                             I       I
I I                                                        I                                             I       I
I I                                                        I                                             I       I
I I                  ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I
I I                  •      IF( NPRINT .LT. 5 )                                                   •......I.......I.......O
I I                  •     * GO TO 370                                                            •      I       I       I
I I                  ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I       I
I I                                                        I                                             I       I       I
I I                                                        I                                             I       I       I
I I                                                        I                                             I       I       I
I I                  ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I       I
I I                  •      WRITE( 6, NOISES )                                                    •      I       I       I
I I                  ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I       I
I I                                                        I                                             I       I       I
I I                                                        I                                             I       I       I
I I                                                        I                                             I       I       I
I I                  ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I       I
I I                  •      IF( NPRINT .LE. 7 )                                                   •......I.......I......)V
I I                  •     * GO TO 370                                                            •      I       I       I
I I                  ••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I       I
I I                                                        I                                             I       I       I
I I                                                        I                                             I       I       I
```



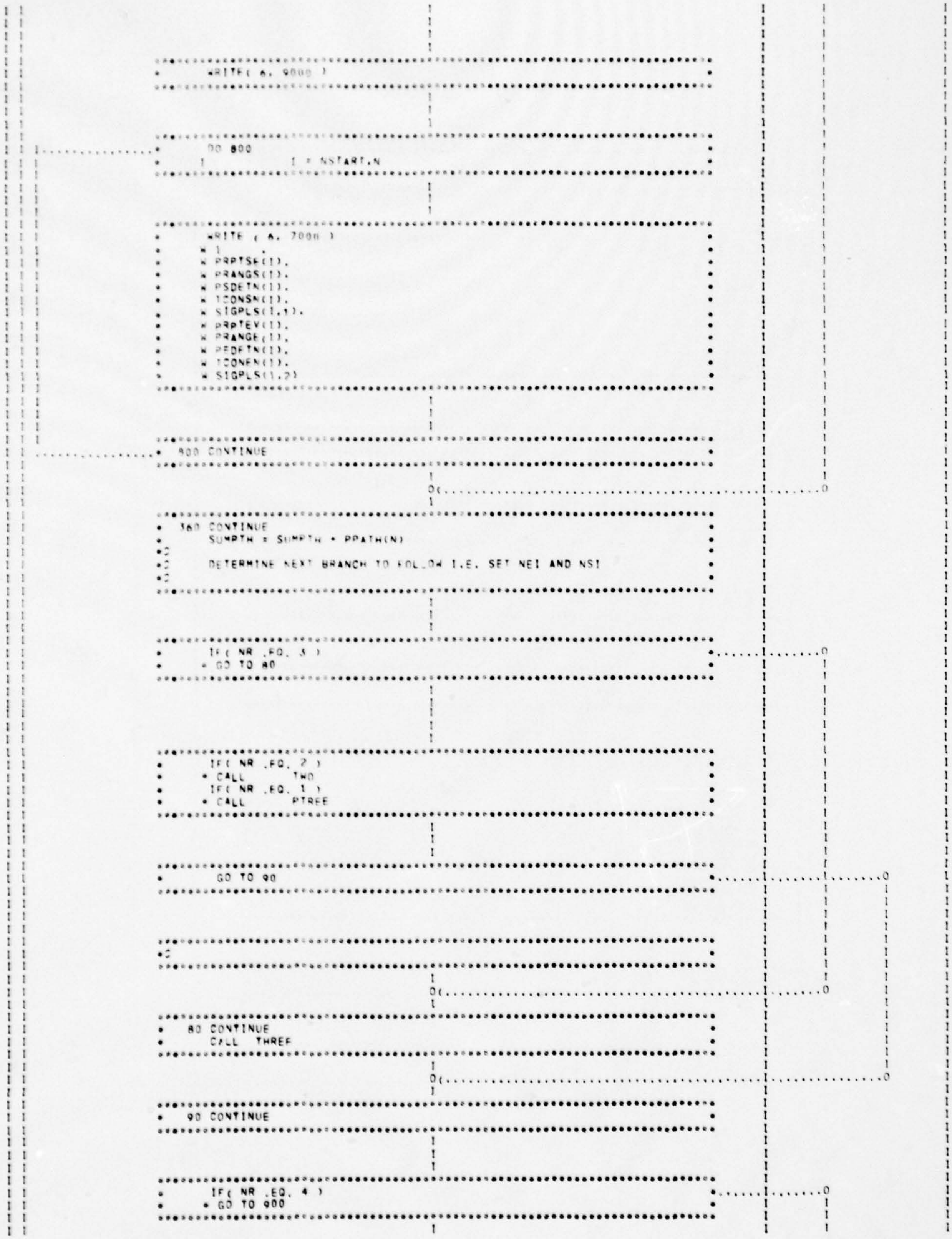
```
      WRITE( 6, BOTTOM )

  370 CONTINUE
      STNPSE = SIGPLS(N,1)
      STNPEV = SIGPLS(N,2)
      NTIMEN = N
C
C     COMPUTE   MEAN AND VARIANCE OF POWER VALUES
C
      CALL
     S            MUEVAR
C
C     COMPUTE   PROBABILITY OF DETECTION FOR THIS TIME
C
      CALL
     S            PRODET

   40 CONTINUE
C
C     COMPUTE   THE D- AND E- STATES
C
      CALL
     S            STAT
C
C     SET  UP REQUIRED PARTS OF D- AND E- STATE TABLES
C
      CALL
     S            TABLE
C
C     COMPUTE   THE PROBABILITIES OF EVADE, KILL, AND PATH
C
      CALL
     S            PROBL
C
C     MODIFY   THE MEAN AND VARIANCES OF THE POWERS
C
      CALL
     S            DEMUVA
     S ( NDSTAT, NESTAT )
C
C     COMPUTE  PROBABILITIES MODIFIED BY PATH PROBABILITY
C

      IF( N .EQ. 1 )
     * GO TO 50

      CALL
     S            PROBAL

   50 CONTINUE
C
C     ACCUMULATE   THE PROBABILITIES
C
      CALL
     S            ACCSTA
C
C     DETERMINE IF BRANCH SHOULD BE TERMINATED
C
```



```
      IF( NTIMEN .GE. MAXTIM )
     * GO TO 70

      IF( (PPATH(N) .LT. PPAMIN) .OR. (( (N .GT. 1) .AND. (RANGE(N) .LT.
     * RANGE(N-1)) ) .AND. ( (PROSER(N) .LT. PDSMIN) .AND. (PROEVA(N)
     * .LT. PDEMIN) )) )
     * GO TO 70

C
C     UPDATE QUANTITIES FOR NEXT POINT ON BRANCH
C
      PPATHM = PPATH(N)
      N = N + 1

      IF( (NPCO+MECO .EQ. 9) .AND. (NR .GT. 1) )
     * GO TO 370

C
C     UPDATE   SHIP MOTION AND POSITION
C
      CALL
     S           UPDATE
      CALL
     S           UPPOS

       GO TO 30

C

   70 CONTINUE

      IF( NPRINT .LT. 4 )
     * GO TO 360

      BRNKIL = SUMKIL - BRNKIL
      BRNEVA = SUMEVA - BRNEVA
      AVBRKL = BRNKIL/AMAXNN
      AVBRFV = BRNEVA/AMAXNN
      WRITE( 6, BRANCH )
      BRNKIL = SUMKIL
      BRNEVA = SUMEVA

      IF( NPRINT .LT. 6 )
     * GO TO 360

      WRITE ( 6, 8000 )
```



```
      DO 600
      I              I = NSTART,N

      WRITE ( 6, 7000 )
     W I        .
     W PXS    (I).
     W PYS    (I).
     W RANGE  (I).
     W PXE    (I).
     W PYE    (I).
     W PKILL  (I).
     W PEVADE(I).
     W PPATH  (I)

  600 CONTINUE

      IF( NPRINT .LT. 7 )
     * GO TO 360

      WRITE ( 6, 6000 )

      DO 700
      I              I = NSTART,N

      WRITE ( 6, 7000 )
     W I        .
     W PROSER(I).
     W GMUSER(I).
     W VARSER(I).
     W DEMUSE(I).
     W DEVASE(I).
     W PROEVA(I).
     W GMUEVA(I).
     W VAREVA(I).
     W DEMUEV(I).
     W DEVAEV(I)

  700 CONTINUE

      IF( NPRINT .LT. 8 )
     * GO TO 360
```



```
      WRITE( 6, 9000 )

      DO 800
     1             I = NSTART,N

      WRITE ( 6, 7000 )
     W 1     ,
     W PRPTSE(I),
     W PRANGS(I),
     W PSDETN(I),
     W ICONSN(I),
     W SIGPLS(I,1),
     W PRPTEV(I),
     W PRANGE(I),
     W PEDETN(I),
     W ICONEN(I),
     W SIGPLS(I,2)

  800 CONTINUE

  360 CONTINUE
      SUMPTH = SUMPTH + PPATH(N)
C
      DETERMINE NEXT BRANCH TO FOLLOW I.E. SET NEI AND NSI
C

      IF( NR .EQ. 3 )
     * GO TO 80

      IF( NR .EQ. 2 )
     * CALL      TWO
      IF( NR .EQ. 1 )
     * CALL      PTREE

       GO TO 90

C

   80 CONTINUE
      CALL  THREE

   90 CONTINUE

      IF( NR .EQ. 4 )
     * GO TO 900
```

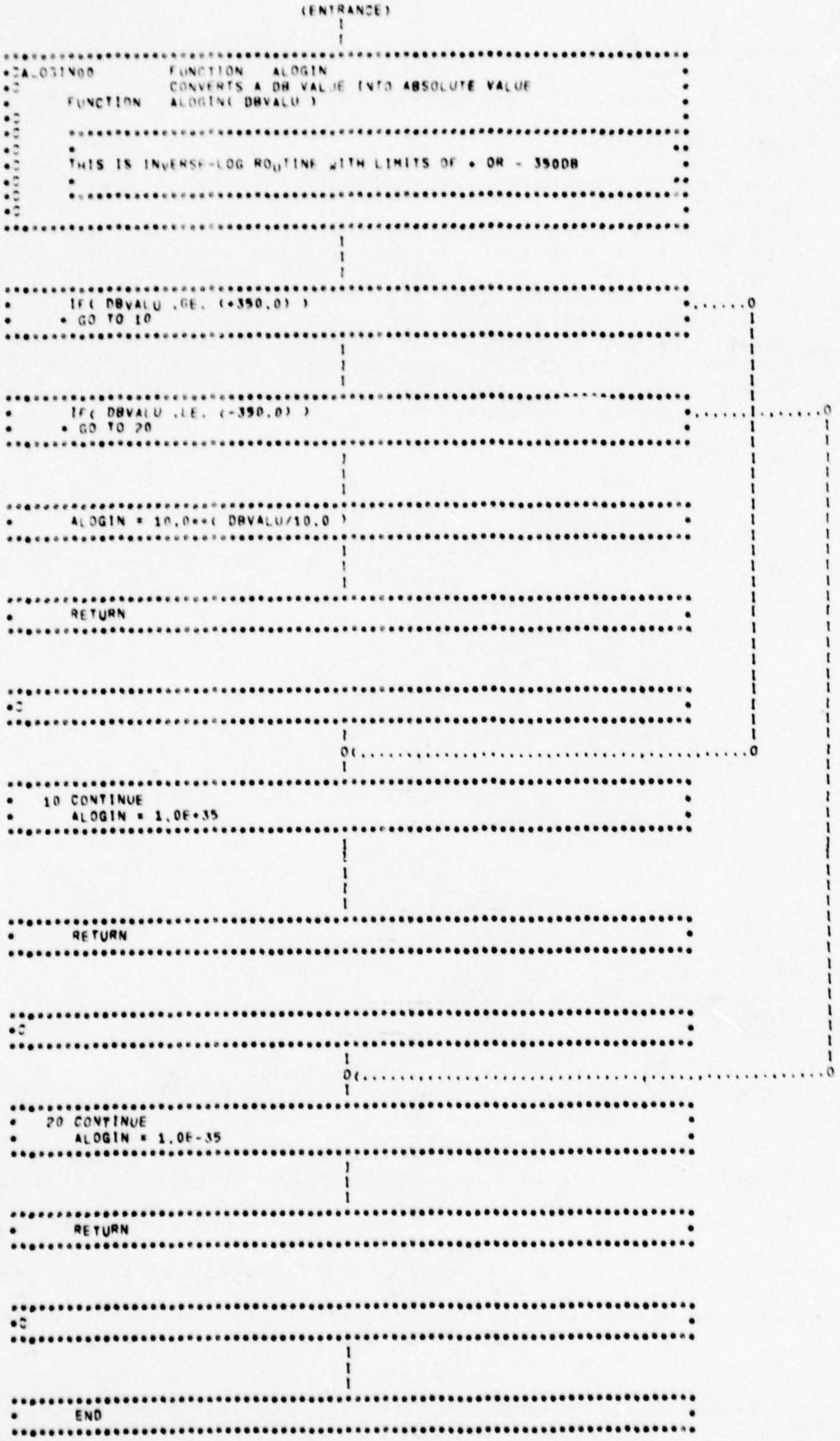
```
I I                                                   I                                           I       I               I
I I                                                   I                                           I       I               I
I I                •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I               I
I I                •C                                                                      •      I       I               I
I I                •C      RESET   CLOSE AND EVADE PHASES                                  •      I       I               I
I I                •C                                                                      •      I       I               I
I I                •       CALL                                                            •      I       I               I
I I                •      S              RESET                                             •      I       I               I
I I                •       N = MAXO( NSI, NEI ) - 1                                        •      I       I               I
I I                •       NSTART = N                                                      •      I       I               I
I I                •       IF( N .GT. NSI )                                                •      I       I               I
I I                •      * CLPH = 1.0                                                     •      I       I               I
I I                •       IF( N .GT. NEI )                                                •      I       I               I
I I                •      * EVPH = 1.0                                                     •      I       I               I
I I                •       PPATHM = PPATH(N-1)                                             •      I       I               I
I I                •       IF( N .EQ. 1 )  PPATHM = 1.0                                    •      I       I               I
I I                •       NTIMEN = N                                                      •      I       I               I
I I                •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I               I
I I                                                   I                                           I       I               I


I I                                                   I                                           I       I               I
I I                                                   I                                           I       I               I
I I                                                   I                                           I       I               I
I I                                                   I                                           I       I               I
I I                •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I               I
I I                •       GO TO 40                                                        •......I.......I...............0
I I                •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I
I I                                                                                               I       I
I I                                                                                               I       I
I I                                                                                               I       I
I I                •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I
I I                •C                                                                             I       I
I I                •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I
I I                                                   I                                           I       I
I I                                                   0(..........................................I.......0
I I                                                   I                                           I
I  ................•  900 CONTINUE                                                          •      I
I                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                                                     I                                           I
I                                                     I                                           I
I                                                     I                                           I
I                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                  •       HE1 = HE1 + DELHED                                              •      I
I                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                                                     I                                           I
I                                                     I                                           I
I                                                     I                                           I
I                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                  •       IF( NPRINT .LE. 8 )                                             •......I.......0
I                  •      * GO TO 350                                                      •      I       I
I                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I
I                                                     I                                           I       I
I                                                     I                                           I       I
I                                                     I                                           I       I
I                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I
I                  •       WRITE( 6. TABLES )                                              •      I       I
I                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I       I
I                                                     I                                           I       I
I                                                     0(..........................................I.......0
I                                                     I                                           I
I                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                  •  350 CONTINUE                                                         •      I
I                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                                                     I                                           I
I                                                     I                                           I
I                                                     I                                           I
I                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
...................•  110 CONTINUE                                                         •      I
                   •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
                                                      I                                           I
                                                      I                                           I
                                                      I                                           I
                   •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
                   •C                                                                      •      I
                   •C      NORMALIZE   AND PRINT   OUTPUT                                  •      I
                   •C                                                                      •      I
                   •       CONNOR = MAXTIM*NEMAX*NSMAX*NUMHED*NUMRNG                       •      I
                   •       AVEEVA = SUMEVA/CONNOR                                          •      I
                   •       AVEKIL = SUMKIL/CONNOR                                          •      I
                   •       AVEPTH = SUMPTH/CONNOR*AMAXNN                                   •      I
                   •       TRENOR = NUMRNG*NUMHED                                          •      I
                   •       SMRANG = SMRANG/TRENOR                                          •      I
                   •       SMPEDT = SMPEDT/TRENOR                                          •      I
                   •       SMPSDT = SMPSDT/TRENOR                                          •      I
                   •       SMTCNE = SMTCNE/TRENOR                                          •      I
                   •       SMTCNS = SMTCNS/TRENOR                                          •      I
                   •       TREKIL = SUMKIL/TRENOR                                          •      I
                   •       TREEVA = SUMEVA/TRENOR                                          •      I
                   •       TREPTH = SUMPTH/TRENOR                                          •      I
                   •       SUMTOT = TREEVA + TREKIL + TREPTH                               •      I
                   •       PSRANG = PSRANG/TRENOR                                          •      I
                   •       PERANG = PERANG/TRENOR                                          •      I
                   •        WRITE ( 6.12000 )                                              •      I
```

```
*   W TREKIL,                                          *    I
*   W TREFVA,                                          *    I
*   W SMPEDT,                                          *    I
*   W SMPSDT                                           *    I
*    WRITE ( 6,13000 )                                 *    I
*   W PERANG,                                          *    I
*   W PSRANG,                                          *    I
*   W SMTCNE,                                          *    I
*   W SMTCNS,                                          *    I
*   W TRENOR,                                          *    I
*   W SUMTOT                                           *    I
*    WRITE ( 6, 1000 )                                 *    I
********************************************************    I
                           I                                I
                           I                                I
                           I                                I
********************************************************    I
*     IF( NPRINT .LE. 0 )                              *....I.....0
*   * GO TO 340                                        *    I     I
********************************************************    I     I
                           I                                I     I
                           I                                I     I
                           I                                I     I
********************************************************    I     I
*C                                                     *    I     I
*C   PRINT OUT DETAILED LIST OF INPUTS                      I     I
*C                                                     *    I     I
*    WRITE ( 6,21000 )                                 *    I     I
*   W ACZ   ,                                          *    I     I
*   W ALPHAE,                                          *    I     I
*   W ALPHAS,                                          *    I     I
*   W AR1X  ,                                          *    I     I
*   W AR1Y  ,                                          *    I     I
*   W AR1Z  ,                                          *    I     I
*   W AR2X  ,                                          *    I     I
*   W AR2Y  ,                                          *    I     I
*   W AR2Z                                             *    I     I
*    WRITE ( 6,22000 )                                 *    I     I
*   W B2E   ,                                          *    I     I
*   W B2S   ,                                          *    I     I
*   W BCZ   ,                                          *    I     I
*   W DELANG,                                          *    I     I
*   W DELF  ,                                          *    I     I
*   W DELHED,                                          *    I     I
*   W DELRNG,                                          *    I     I
*   W DIFTI                                            *    I     I
*    WRITE ( 6,23000 )                                 *    I     I
*   W DPFTEV,                                          *    I     I
*   W DPFTSE,                                          *    I     I
*   W DTRAD ,                                          *    I     I
*   W F11   ,                                          *    I     I
*   W E12   ,                                          *    I     I
*   W E13   ,                                          *    I     I
*   W E21   ,                                          *    I     I
*   W F22   ,                                          *    I     I
*   W E23                                              *    I     I
*    WRITE ( 6,24000 )                                 *    I     I
*   W FOE   ,                                          *    I     I
*   W FOS   ,                                          *    I     I
*   W F1E   ,                                          *    I     I
*   W F1S   ,                                          *    I     I
*   W F2E   ,                                          *    I     I
*   W F2S   ,                                          *    I     I
*   W FBWE  ,                                          *    I     I
*   W FBWS                                             *    I     I
*    WRITE ( 6,25000 )                                 *    I     I
*   W FRES1 ,                                          *    I     I
*   W FRES2 ,                                          *    I     I
*   W HDINEV,                                          *    I     I
*   W HEDMAX,                                          *    I     I
*   W HS1   ,                                          *    I     I
*   W I01   ,                                          *    I     I
*   W I02   ,                                          *    I     I
*     W MAXLAY,                                                             *      I       I
*     W MAXTIM                                                              *      I       I
*       WRITE ( 6,26000 )                                                   *      I       I
*     W MECO   ,                                                            *      I       I
*     W NEMAX  ,                                                            *      I       I
*     W NEWLAY,                                                             *      I       I
*     W NPCO   ,                                                            *      I       I
*     W NPRINT,                                                             *      I       I
*     W NSMAX  ,                                                            *      I       I
*     W NULAPO,                                                             *      I       I
*     W PDEMIN,                                                             *      I       I
*     W PDSMIN                                                              *      I       I
*       WRITE ( 6,27000 )                                                   *      I       I
*     W POR    ,                                                            *      I       I
*     W PPAMIN,                                                             *      I       I
*     W PRE    ,                                                            *      I       I
*     W PRK    ,                                                            *      I       I
*     W QTRAN1,                                                             *      I       I
*     W QTRAN2,                                                             *      I       I
*     W RGINEV,                                                             *      I       I
*     W RI     ,                                                            *      I       I
*     W RNGMAX                                                              *      I       I
*       WRITE ( 6,28000 )                                                   *      I       I
*     W SPDEVA,                                                             *      I       I
*     W SPDSER,                                                             *      I       I
*     W SS     ,                                                            *      I       I
*     W THREVA,                                                             *      I       I
*     W THRSER,                                                             *      I       I
*     W WRANGE                                                              *      I       I
*21000 FORMAT(                                                              *      I       I
*     F / F15.6, 99H = ACZ    = FIRST CONVERGENCE ZONE CONSTANT             *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = ALPHAE = SMOOTHING PARAMETER FOR EVADER              *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = ALPHAS = SMOOTHING PARAMETER FOR SEARCHER            *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = AR1X   = X-PRIME ARRAY DIMENSION FOR SEARCHER        *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = AR1Y   = Y-PRIME ARRAY DIMENSION FOR SEARCHER        *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = AR1Z   = Z-PRIME ARRAY DIMENSION FOR SEARCHER        *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = AR2X   = X-PRIME ARRAY DIMENSION FOR EVADER          *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = AR2Y   = Y-PRIME ARRAY DIMENSION FOR EVADER          *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = AR2Z   = Z-PRIME ARRAY DIMENSION FOR EVADER          *      I       I
*     F                                                                     *      I       I
*     F )                                                                   *      I       I
*22000 FORMAT(                                                              *      I       I
*     F / F15.6, 99H = B2E    = INTEGRATION BANDWIDTH OF POST-DETECTION    O*      I       I
*     FF EVADER                                                             *      I       I
*     F / F15.6, 99H = B2S    = INTEGRATION BANDWIDTH OF POST-DETECTION    O*      I       I
*     FF SEARCHER                                                           *      I       I
*     F / F15.6, 99H = BCZ    = SECOND CONVERGENCE ZONE CONSTANT            *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = DELANG = ANGLE STEP SIZE FOR RAY PRE-TRACE TABLE     *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = DELF   = FREQUENCY INCREMENT                         *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = DELHED = CHANGE IN HEADING OF EVADER                 *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = DELRNG = CHANGE IN RANGE OF CLOSEST APPROACH         *      I       I
*     F                                                                     *      I       I
*     F / F15.6, 99H = DIFTI  = TIME BETWEEN POINTS ON A BRANCH             *      I       I
*     F                                                                     *      I       I
*     F )                                                                   *      I       I
```

```
*23000 FORMAT(                                                    *        I         I
*      F / F15.6, 99H = DPFTEV = EVADER DEPTH                     *        I         I
*      F                                                          *        I         I
*      F / F15.6, 99H = DPFTSE = SEARCHER DEPTH                   *        I         I
*      F                                                          *        I         I
*      F / F15.6, 99H = DTRAD  = ANGLE INCREMENT FOR RADIATED SIGNAL      *        I         I
*      F                                                          *        I         I
*      F / F15.6, 99H = E11    = FIRST  EULER ANGLE FOR SEARCHER (Z AXIS R*        I         I
*      FOTATION)                                                  *        I         I
*      F / F15.6, 99H = E12    = SECOND EULER ANGLE FOR SEARCHER (LINE OF *        I         I
*      FNODES ROTATION)                                           *        I         I
*      F / F15.6, 99H = E13    = THIRD  EULER ANGLE FOR SEARCHER (Z-PRIME *        I         I
*      FAXIS ROTATION)                                            *        I         I
*      F / F15.6, 99H = E21    = FIRST  EULER ANGLE FOR  EVADER  (Z AXIS R*        I         I
*      FOTATION)                                                  *        I         I
*      F / F15.6, 99H = E22    = SECOND EULER ANGLE FOR  EVADER  (LINE OF *        I         I
*      FNODES ROTATION)                                           *        I         I
*      F / F15.6, 99H = E23    = THIRD  EULER ANGLE FOR  EVADER  (Z-PRIME *        I         I
*      FAXIS ROTATION)                                            *        I         I
*      F )                                                        *        I         I
*24000 FORMAT(                                                    *        I         I
*      F / F15.6, 99H = F0E    = PRE-DETECTION FILTER CENTER FREQUENCY FOR*        I         I
*      F EVADER                                                   *        I         I
*      F / F15.6, 99H = F0S    = PRE-DETECTION FILTER CENTER FREQUENCY FOR*        I         I
*      F SEARCHER                                                 *        I         I
*      F / F15.6, 99H = F1E    = LOWER FREQUENCY LIMIT OF EVADER EQUIPMENT*        I         I
*      F                                                          *        I         I
*      F / F15.6, 99H = F1S    = LOWER FREQUENCY LIMIT OF SEARCHER EQUIPME*        I         I
*      FNT                                                        *        I         I
*      F / F15.6, 99H = F2E    = UPPER FREQUENCY LIMIT OF EVADER EQUIPMENT*        I         I
*      F                                                          *        I         I
*      F / F15.6, 99H = F2S    = UPPER FREQUENCY LIMIT OF SEARCHER EQUIPME*        I         I
*      FNT                                                        *        I         I
*      F / F15.6, 99H = FBWE   = PRE-DETECTION FILTER BANDWIDTH OF EVADER *        I         I
*      F                                                          *        I         I
*      F / F15.6, 99H = FBWS   = PRE-DETECTION FILTER BANDWIDTH OF SEARCHE*        I         I
*      FR                                                         *        I         I
*      F )                                                        *        I         I
*25000 FORMAT(                                                    *        I         I
*      F / F15.6, 99H = FRES1  = TRANSDUCER RESONANT FREQUENCY ON SEARCHER*        I         I
*      F                                                          *        I         I
*      F / F15.6, 99H = FRES2  = TRANSDUCER RESONANT FREQUENCY ON EVADER  *        I         I
*      F                                                          *        I         I
*      F / F15.6, 99H = HDINEV = INITIAL HEADING OF EVADER        *        I         I
*      F                                                          *        I         I
*      F / F15.6, 99H = HEDMAX = MAXIMUM HEADING OF EVADER        *        I         I
*      F                                                          *        I         I
*      F / F15.6, 99H = HS1    = HEADING OF SEARCHER              *        I         I
*      F                                                          *        I         I
*      F / I15,   99H = I01    = NUMBER OF DIRECTIVITY ANGLES FOR SEARCHER*        I         I
*      F                                                          *        I         I
*      F / I15,   99H = I02    = NUMBER OF DIRECTIVITY ANGLES FOR EVADER  *        I         I
*      F                                                          *        I         I
*      F / I15,   99H = MAXLAY = MAXIMUM NUMBER OF LAYERS TO BE USED TO FI*        I         I
*      FT PROFILE                                                 *        I         I
*      F / I15,   99H = MAXTIM = MAXIMUM NUMBER OF POINTS ON A BRANCH     *        I         I
*      F                                                          *        I         I
*      F )                                                        *        I         I
*26000 FORMAT(                                                    *        I         I
*      F / I15,   99H = MECO   = CONTROL PARAMETER FOR EVASION COURSE OPTI*        I         I
*      FONS                                                       *        I         I
*      F / I15,   99H = NEMAX  = MAXIMUM POINT ALONG BRANCH TO START EVASI*        I         I
*      FON                                                        *        I         I
*      F / I15,   99H = NEWLAY = CONTROL CONSTANT FOR INITIALIZATION      *        I         I
*      F                                                          *        I         I
*      F / I15,   99H = NPCO   = CONTROL PARAMETER FOR PURSUIT COURSE OPTI*        I         I
*      FON                                                        *        I         I
*      F / I15,   99H = NPRINT = CONTROL PARAMETER FOR AMOUNT OF PRINTOUT *        I         I
*      F                                                          *        I         I
*      F / I15,   99H = NSMAX  = MAXIMUM POINT ALONG BRANCH TO START CLOSI*        I         I
*      FNG                                                        *        I         I
*      F / I15,   99H = NULAPO = NUMBER OF LAYERS PLUS ONE        *        I         I
*      F                                                          *        I         I
*     F / F15.6, 99H = PDEMIN = MINIMUM DETECTION PROBABILITY ALLOWED AFT*      I       I
*     FER SHIPS START SEPARATING AS SEEN BY EVADER                       *      I       I
*     F / F15.6, 99H = PDSMIN = MINIMUM DETECTION PROBABILITY ALLOWED AFT*      I       I
*     FER SHIPS START SEPARATING AS SEEN BY SEARCHER                     *      I       I
*     F )                                                                *      I       I
*27000 FORMAT(                                                           *      I       I
*     F / F15.6, 99H = POR    = POROSITY OF BOTTOM                       *      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = PPAMIN = MINIMUM PATH PROBABILITY TO BE CONSIDERED*      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = PRE    = A-PRIORI PROBABILITY OF EVASION          *      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = PRK    = A-PRIORI PROBABILITY OF A KILL           *      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = QTRAN1 = TRANSDUCER FIGURE OF MERIT ON SEARCHER   *      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = QTRAN2 = TRANSDUCER FIGURE OF MERIT ON EVADER     *      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = RGINEV = INITIAL RANGE OF CLOSE                   *      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = RI     = INITIAL RANGE BETEEN SHIPS AT START OF BR*      I       I
*     FANCH                                                              *      I       I
*     F / F15.6, 99H = RNGMAX = MAXIMUM CLOSEST APPROACH DISTANCE        *      I       I
*     F                                                                  *      I       I
*     W )                                                                *      I       I
*28000 FORMAT(                                                           *      I       I
*     F / F15.6, 99H = SPDEVA = SPEED OF EVADER                          *      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = SPDSER = SPEED OF SEARCHER                        *      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = SS     = SEA STATE                                *      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = THREVA = DETECTION THRESHOLD FOR THE EVADER       *      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = THRSER = DETECTION THRESHOLD FOR THE SEARCHER     *      I       I
*     F                                                                  *      I       I
*     F / F15.6, 99H = WRANGE = WEAPON RANGE                             *      I       I
*     F                                                                  *      I       I
*     W )                                                                *      I       I
********************************************************************      I       I
                                  I                                       I       I
                                  I                                       I       I
                                  I                                       I       I
********************************************************************      I       I
*       IF( NPRINT .LE. 9 )                                            *......I......)V
*       * GO TO 340                                                    *      I       I
********************************************************************      I       I
                                  I                                       I       I
                                  I                                       I       I
                                  I                                       I       I
********************************************************************      I       I
*       WRITE( 6, POWERS )                                             *      I       I
*       WRITE( 6, EVALUE )                                             *      I       I
*       WRITE( 6, DATAOU )                                             *      I       I
*       CALL  PDUMP                                                    *      I       I
********************************************************************      I       I
                                  I                                       I       I
                                  O(......................................I.......O
                                  I                                       I
********************************************************************      I
*   340 CONTINUE                                                       *      I
*       NEWLAY = 1                                                     *      I
********************************************************************      I
                                  I                                       I
                                  I                                       I
                                  I                                       I
********************************************************************      I
*       GO TO 10                                                       *......O
********************************************************************
```

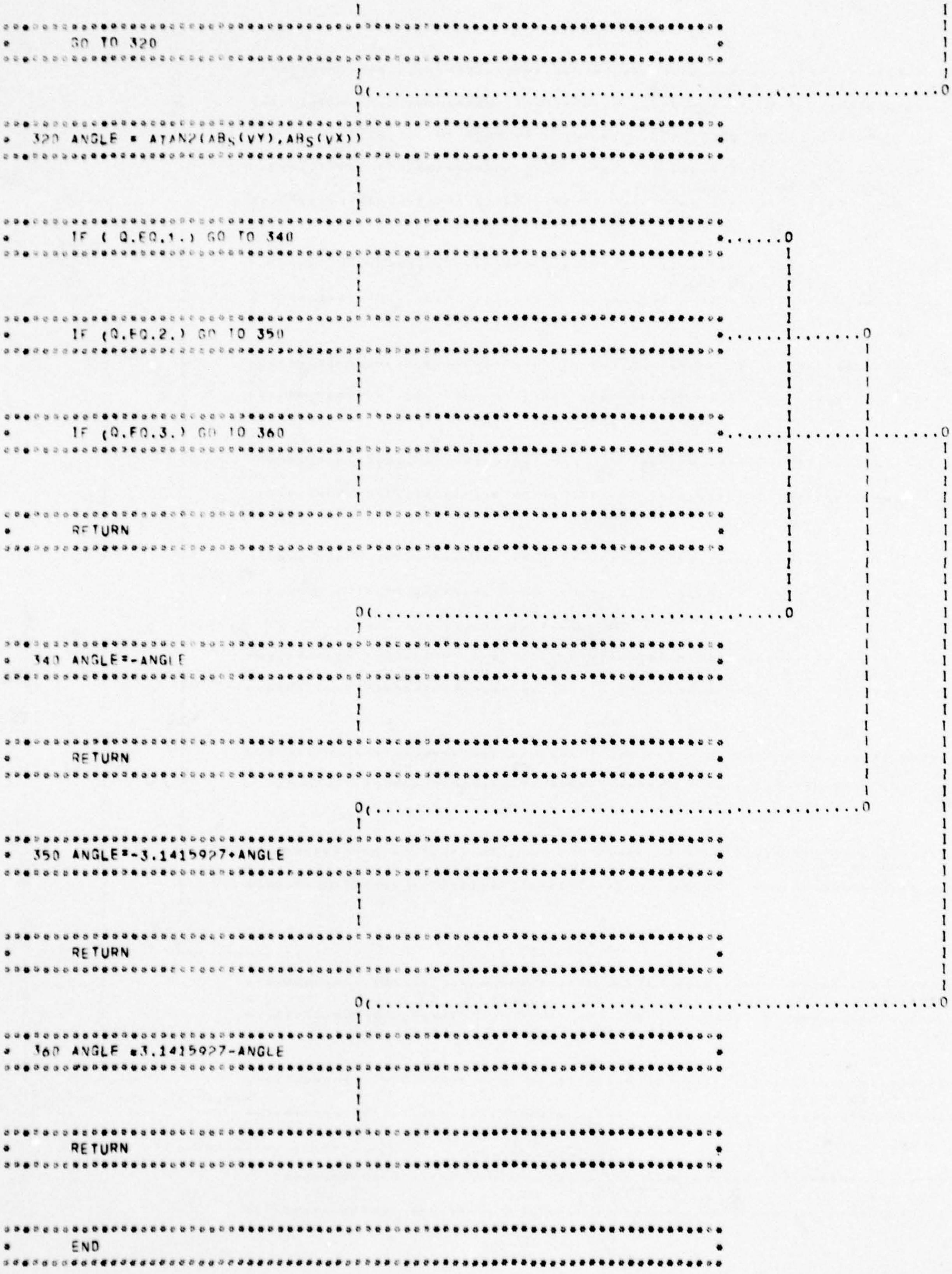
```
*********************************************************************
*C                                                                  *
* 1000 FORMAT(                                                      *
*      F 1H1 )                                                      *
* 2000 FORMAT(                                                      *
*      F    58H1LAYER  -DEPTH    -DELTA-Z  -VELOCITY          -VO       -G0 *
*      F   59H               -G1          -G2      -SLOPE  -MIDSPEED        )*
* 3000 FORMAT(                                                      *
*      F 6E20.10 )                                                  *
* 4000 FORMAT(                                                      *
*      F 52H1RADIAN ANGLE       RANGES FOR DIFFERENT BOUNCE MODES  /  *
*      F 24X, 90H DIRECT        ONE-SURFACE       ONE-BOTTOM   SURFACE-BOTTOM *
*      F  BOTTOM-SURFACE        TWO-SURFACE )                       *
* 5000 FORMAT(                                                      *
*      F 7E17.7 )                                                   *
* 6000 FORMAT(                                                      *
*      F  //59H N  PR-SEARCHER        MU-S     SIGMA-S     DEL-MU  DEL-SIGMA *
*      F   59H        PR-EVADER       MU-E     SIGMA-E     DEL-MU  DEL-SIGMA)*
* 7000 FORMAT(                                                      *
*      F I3, 1P5E11.3, 5X, 5F11.3 )                                 *
* 8000 FORMAT(                                                      *
*      F  / 58H N   SEARCHER-X          -Y        RANGE   EVADER-X         -Y *
*      F    59H PROBABIL.-KILL      -EVADE        -PATH                    )*
* 9000 FORMAT(                                                      *
*      F /  58H N         S-PATH     -RANGE -DETECTION       -TIME      -S/N+1 *
*      F    59H           E-PATH     -RANGE -DETECTION       -TIME      -S/N+1)*
*10000 FORMAT(                                                      *
*      F 51H COLLISION PURSUIT AND NORMAL ESCAPE NOT COMPATIBLE  )  *
*11000 FORMAT(                                                      *
*      F 60H COLLISION COURSE PURSUIT WILL NOT WORK WITH SLOWER SEARCHER )*
*12000 FORMAT(                                                      *
*      F 1H1                                                        *
*      F / F15.6, 94H   = AVERAGE PROBABILITY OF SEARCHER NEUTRALIZING EVA*
*      FDER                                                         *
*      F / F15.6, 94H   = AVERAGE PROBABILITY OF EVADER ESCAPING FROM SEAR*
*      FCHER                                                        *
*      F / F15.6, 94H   = AVERAGE PROBABILITY OF FIRST DETECTION FOR EVADE*
*      FR                                                           *
*      F / F15.6, 94H   = AVERAGE PROBABILITY OF FIRST DETECTION FOR SEARC*
*      FHER                                                         *
*      F )                                                          *
*13000 FORMAT(                                                      *
*      F / F15.6, 94H   = AVERAGE RANGE OF FIRST DETECTION FOR EVADER (KYD*
*      F)                                                           *
*      F / F15.6, 94H   = AVERAGE RANGE OF FIRST DETECTION FOR SEARCHER (K*
*      FYD)                                                         *
*      F / F15.6, 94H   = AVERAGE LENGTH OF TIME DURING WHICH EVADER IS DE*
*      FTECTING SEARCHER PER TIME INTERVAL (SEC)                    *
*      F / F15.6, 94H   = AVERAGE LENGTH OF TIME DURING WHICH SEARCHER IS *
*      FDETECTING EVADER PER TIME INTERVAL (SEC)                    *
*      F / F15.6, 94H   = NUMBER OF COMBINATIONS OF INITIAL HEADINGS AND C*
*      FLOSEST APPROACH DISTANCES                                   *
*      F / F15.6, 94H   = SUM OF KILL, EVADE AND RESIDUE PATH PROBABILITIE*
*      FS                                                           *
*      F 1H1 )                                                      *
*C                                                                  *
*********************************************************************
                                  I
                                  I
                                  I
*********************************************************************
*      END                                                          *
*********************************************************************
```

```
                              (ENTRANCE)
                                  I
                                  I
****************************************************************************
*CACCV        ACCUMULATIVE STATISTICS                                      *
*C   ***********************************************************************
*C   *                                                                     *
*C   *  ACCUM. STATISTICS                                                  *
*C   *                                                                     *
*C   ***********************************************************************
*      SUBROUTINE ACCSTA                                                   *
*      COMMON /LAREL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*     1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NS1,*
*     2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*     3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*     4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*     5LSUBE,ALSUBS,SINPSE,SINPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*     6,F0S,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*     7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*     8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                    *
*      COMMON                                                              *
*     C / STATIC /                                                         *
*     C            NOSTAT,                                                 *
*     C            NESTAT,                                                 *
*     D            PEDETN(128)        ,                                    *
*     C            PERANG,                                                 *
*     C            PPATHM,                                                 *
*     D            PRANGE(128)        ,                                    *
*     D            PRANGS(128)        ,                                    *
*     D            PRPTEV(128)        ,                                    *
*     D            PRPTSE(128)        ,                                    *
*     D            PSDETN(128)        ,                                    *
*     C            PSRANG,                                                 *
*     C            SMPEDT,                                                 *
*     C            SMPSDT,                                                 *
*     C            SMTCNE,                                                 *
*     C            SMTCNS,                                                 *
*     D            TCONEN(128)                                             *
*     D            TCONSN(128)                                             *
*     CDUMMYB                                                              *
*C                                                                         *
*      PERANG = PERANG + PRANGE(N)                                         *
*      PSRANG = PSRANG + PRANGS(N)                                         *
*      SMPSDT = SMPSDT + PSDETN(N)                                         *
*      SMPEDT = SMPEDT + PEDETN(N)                                         *
*      SMTCNE = SMTCNE + TCONEN(N)                                         *
*      SMTCNS = SMTCNS + TCONSN(N)                                         *
*      SUMEVA = SUMEVA + PEVADE(N)                                         *
*      SUMKIL = SUMKIL + PKILL(N)                                          *
****************************************************************************
                                  I
                                  I
                                  I
****************************************************************************
*      RETURN                                                              *
****************************************************************************

****************************************************************************
*C                                                                         *
****************************************************************************
                                  I
                                  I
                                  I
****************************************************************************
*     END                                                                  *
****************************************************************************
```



```
(ENTRANCE)

CALOGIN00        FUNCTION   ALOGIN
C                CONVERTS A DB VALUE INTO ABSOLUTE VALUE
      FUNCTION   ALOGIN( DBVALU )
C
C
C     THIS IS INVERSE-LOG ROUTINE WITH LIMITS OF + OR - 350DB
C
C
C

      IF( DBVALU .GE. (+350.0) )
     * GO TO 10

      IF( DBVALU .LE. (-350.0) )
     * GO TO 20

      ALOGIN = 10.0**( DBVALU/10.0 )

      RETURN

C

   10 CONTINUE
      ALOGIN = 1.0E+35

      RETURN

C

   20 CONTINUE
      ALOGIN = 1.0E-35

      RETURN

C

      END
```

```
                               (ENTRANCE)
                                    I
                                    I
************************************************************************
*CANGV        TO COMPUTE VECT. ANG.                                    *
*C  ********************************************************************
*C  *                                                                  *
*C  *  COMPUTING ANG OF VECT.V=VXI+VYJ  WITH RESPECT TO  XY AXIS       *
*C  *                                                                  *
*C  ********************************************************************
*      SUBROUTINE ANGVE (VX,VY,ANGLE)                                  *
************************************************************************
                                    I
                                    I
                                    I
************************************************************************
*      IF(VX.LT.0.) GO TO 300                                          *......O
************************************************************************      I
                                    I                                         I
                                    I                                         I
                                    I                                         I
************************************************************************      I
*      IF(VY.LT.0.)GO TO 310                                           *......I.........O
************************************************************************      I         I
                                    I                                         I         I
                                    I                                         I         I
                                    I                                         I         I
************************************************************************      I         I
*      Q=1.                                                            *      I         I
************************************************************************      I         I
                                    I                                         I         I
                                    I                                         I         I
                                    I                                         I         I
************************************************************************      I         I
*      GO TO 320                                                       *......I.........I.........O
************************************************************************      I         I         I
                                   O(.........................................O         I         I
                                    I                                                   I         I
************************************************************************                I         I
*  300 IF(VY.LT.0.) GO TO 330                                          *......O         I         I
************************************************************************      I         I         I
                                    I                                         I         I         I
                                    I                                         I         I         I
                                    I                                         I         I         I
************************************************************************      I         I         I
*      Q=2.                                                            *      I         I         I
************************************************************************      I         I         I
                                    I                                         I         I         I
                                    I                                         I         I         I
                                    I                                         I         I         I
************************************************************************      I         I         I
*      GO TO 320                                                       *......I.........I........)V
************************************************************************      I         I         I
                                    I                                         I         I         I
                                   O(.........................................O         I         I
                                    I                                                   I         I
************************************************************************                I         I
*  330 Q=3.                                                            *                I         I
************************************************************************                I         I
                                    I                                                   I         I
                                    I                                                   I         I
                                    I                                                   I         I
************************************************************************                I         I
*      GO TO 320                                                       *................I........)V
************************************************************************                I         I
                                   O(...................................................O         I
                                    I                                                             I
************************************************************************                          I
*  310 Q=4.                                                            *                          I
************************************************************************                          I
                                    I                                                             I
                                    I                                                             I
```



```
      GO TO 320
  320 ANGLE = ATAN2(ABS(VY),ABS(VX))
      IF ( Q.EQ.1.) GO TO 340
      IF (Q.EQ.2.) GO TO 350
      IF (Q.EQ.3.) GO TO 360
      RETURN
  340 ANGLE=-ANGLE
      RETURN
  350 ANGLE=-3.1415927+ANGLE
      RETURN
  360 ANGLE =3.1415927-ANGLE
      RETURN
      END
```

```
                                (ENTRANCE)
                                    I
                                    I
***************************************************************************
*CARCCOS00        FUNCTION   ARCCOS                                       *
*CFARCC000        FUNCTION FOR ARC-COSINE                                 *
*      FUNCTION   ARCCOS( DUMMY1 )                                        *
*C                                                                        *
*C     ********************************************************************
*C     *                                                                 **
*C     THIS IS THE ARC COSINE FUNCTION                                    *
*C      THE ANSWRR IS IN RADIANS                                          *
*C     THE ANSWER ALWAYS LIES BETWEEN ZERO AND PLUS PI/2                  *
*C     *                                                                 **
*C     ********************************************************************
*C                                                                        *
*C     DUMMY1     = THE ARGUMENT OF THE FUNCTION HOPEFULLY LESS THAN ONE  *
*C                                                                        *
*      ARCCOS = ATAN( SQRT( 1.0/DUMMY1/DUMMY1 - 1.0 ) )                   *
***************************************************************************
                                    I
                                    I
                                    I
***************************************************************************
*      RETURN                                                             *
***************************************************************************


***************************************************************************
*C                                                                        *
***************************************************************************
                                    I
                                    I
                                    I
***************************************************************************
*      END                                                                *
***************************************************************************
```

```
                                (ENTRANCE)
                                    I
                                    I
**************************************************************************
*CASNXOX00        FUNCTION   ASNXOX                                      *
*C                FOR DETERMINING SIN(X)/X                               *
*      FUNCTION   ASNXOX( DUMMY1 )                                       *
*C                                                                       *
*C     *******************************************************************
*C     *                                                                **
*C     FOR DETERMINING THE VALUE OF SINE( X ) OVER X                     *
*C     *                                                                **
*C     *******************************************************************
*C                                                                       *
*      ASNXOX = 1.0                                                      *
*      IF( DUMMY1 .EQ. 0.0 )                                             *
*     *RETURN                                                            *
*      ASNXOX = ( DUMMY1/SIN( DUMMY1 ) )**2                              *
**************************************************************************
                                    I
                                    I
                                    I
**************************************************************************
*      RETURN                                                            *
**************************************************************************



**************************************************************************
*C                                                                       *
**************************************************************************
                                    I
                                    I
                                    I
**************************************************************************
*      END                                                               *
**************************************************************************
```

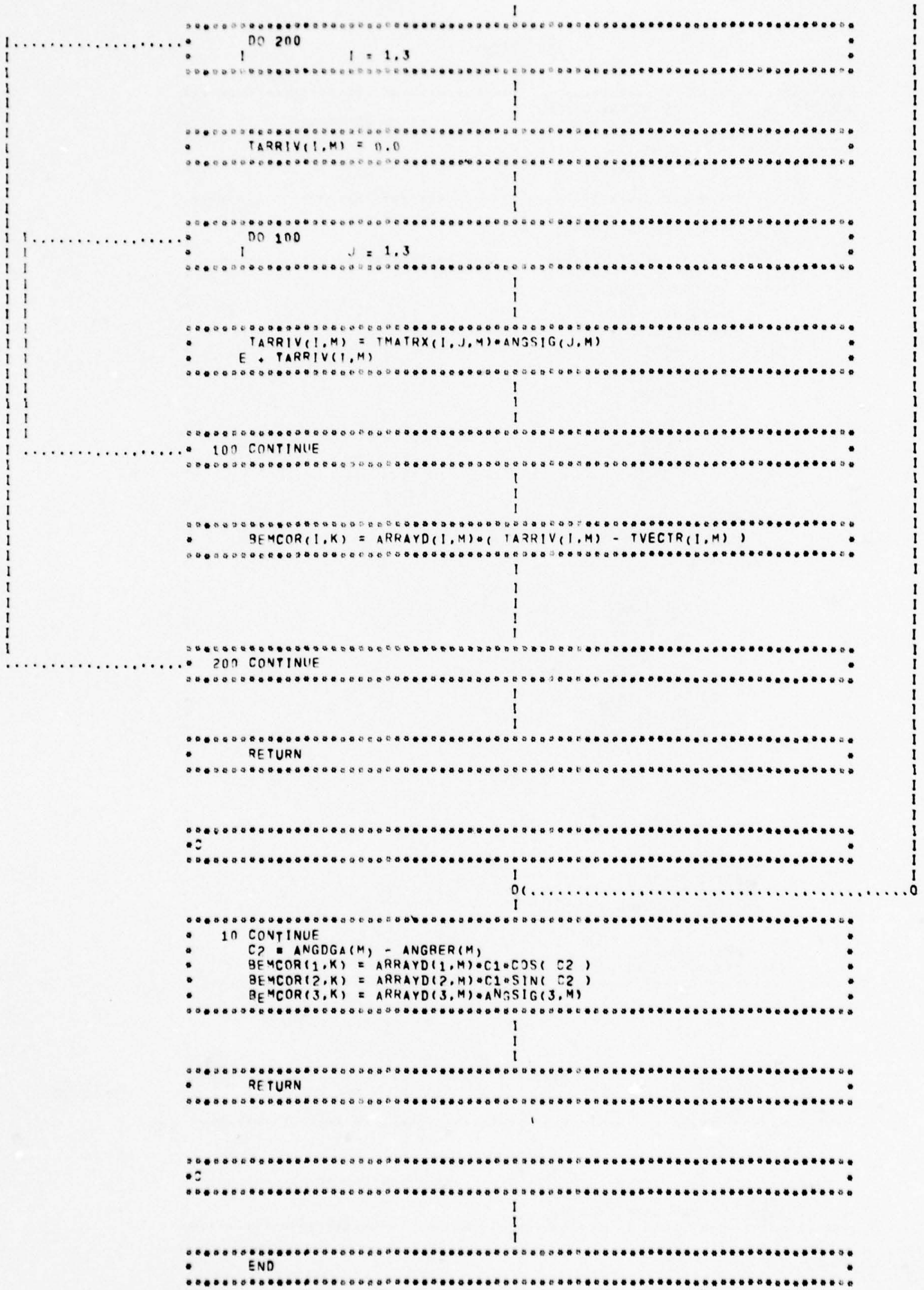
```
                                   (ENTRANCE)
                                       I
                                       I
*************************************************************************
*CBAFFLE00       FUNCTION   BAFFLE                                      *
*C               INTERPOLATES THE BAFFLE AND RADIATED CURVES            *
*      FUNCTION  BAFFLE( N, DUMMY1, X1 )                                *
*      COMMON                                                           *
*     C / SURDUC /                                                      *
*     C          BLA1   ,                                               *
*     C          BLA2   ,                                               *
*     C          BLA3   ,                                               *
*     C          DTRAD  ,                                               *
*     C          J      ,                                               *
*     C          M      ,                                               *
*     D          BAFFUN(128,2)     ,                                    *
*     D          DELBAF(128,2),                                         *
*     D          FLN1(128)         ,                                    *
*     D          FLN2(128)         ,                                    *
*     D          RADSPC(40,50,2)   ,                                    *
*     C          NTIMEN                                                 *
*C                                                                      *
*      X1 = AMOD( DUMMY1, DTRAD )/DTRAD                                 *
*      N = DUMMY1/DTRAD + 1.0                                           *
*      ENTRY BAFCON( N, X1 )                                            *
*      BAFFLE = BAFFUN(N,M) + X1*DE BAF(N,M)                            *
*************************************************************************
                                       I
                                       I
                                       I
*************************************************************************
*      RETURN                                                           *
*************************************************************************


*************************************************************************
*C                                                                      *
*      ENTRY SPETRM( N, DUMMY1, X1 )                                    *
*      X1 = AMOD( DUMMY1, DTRAD )/DTRAD                                 *
*      N = DUMMY1/DTRAD + 1.0                                           *
*      ENTRY OSPECT( N, X1 )                                            *
*      BAFFLE = RADSPC(N,J,M) + X1*( RADSPC(N+1,J,M) - RADSPC(N,J,M) )  *
*************************************************************************
                                       I
                                       I
                                       I
*************************************************************************
*      RETURN                                                           *
*************************************************************************


*************************************************************************
*C                                                                      *
*************************************************************************
                                       I
                                       I
                                       I
                                       I
                                       I
*************************************************************************
*      END                                                              *
*************************************************************************
```

```
                              (ENTRANCE)
                                  I
                                  I
**************************************************************
*CBEAR        RELATIVE BEARING ANGLES AND RANGE              *
*C  ***********************************************************
*C  *                                                        *
*C  *  COMPUTING REL. BEARING ANGLES    AND    RANGE          *
*C  *                                                        *
*C  ***********************************************************
*        SUBROUTINE RELBR                                    *
*        COMMON /LAREL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*       1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*       2NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*       3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*       4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*       5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*       6,F0S,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*       7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*       8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                    *
*        PX=PXE(N)-PXS(N)                                    *
*        PY=PYE(N)-PYS(N)                                    *
*        PROD2=PX**2+PY**2                                   *
*        RANGE(N) = SQRT(PROD2)                              *
*        CBPE = (-VXE(N)*PX-VYE(N)*PY)/(SE1*RANGE(N))        *
*        CBPS = (VXS(N)*PX+VYS(N)*PY)/(SS1*RANGE(N))         *
*        BSP=ACOS(CBPS)                                      *
*        BEP=ACOS(CBPE)                                      *
**************************************************************
                                  I
                                  I
                                  I
**************************************************************
*        RETURN                                              *
**************************************************************


**************************************************************
*        END                                                 *
**************************************************************
```

```
                               (ENTRANCE)
                                   I
                                   I
**************************************************************************
*CBRID       BEARING RIDER                                               *
*C   *********************************************************************
*C   *                                                                   *
*C    *BEARING RIDER                                                     *
*C   *                                                                   *
*C   *********************************************************************
*      SUBROUTINE BRIDER                                                 *
*      COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*     1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NS1,*
*     2NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*     3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*     4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*     5LSUBE,ALSUBS,STNPSE,STNPEV,MICO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*     6,F0S,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*     7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE;P*
*     8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                  *
*      VXS(N-1)=SS1*ALPXN                                                *
*      VYS(N-1)=SS1*ALPYN                                                *
**************************************************************************
                                   I
                                   I
                                   I
**************************************************************************
*      RETURN                                                            *
**************************************************************************



**************************************************************************
*      END                                                               *
**************************************************************************
```

```
                              (ENTRANCE)
                                   I
                                   I
***************************************************************
*CCOLL_I       COLLISION                                      *
*C  ***********************************************************
*C  *                                                         *
*C    *COLLISION                                              *
*C  *                                                         *
*C  ***********************************************************
*       SUBROUTINE COLLIS                                     *
*       COMMON /LAREL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADF(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSURS,SINPSE,SINPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,F0S,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                              *
*       NSUB1=N-1                                                     *
*       AL1= ALPXN*VXF(NSUR1)+ALPYN*VYE(NSUB1)                        *
*       AL=SQRT(AL1**2+SS1**2-SE1**2)-AL1                             *
*       VXS(NSUB1)=VXE(NSUR1)+AL*ALPXN                                *
*       VYS(NSUR1)=VYE(NSUR1)+AL*ALPYN                                *
*********************************************************************
                                   I
                                   I
                                   I
*********************************************************************
*       RETURN                                                      *
*********************************************************************


*********************************************************************
*       END                                                         *
*********************************************************************
```

```
                                (ENTRANCE)
                                    I
                                    I
**************************************************************************
*CCORREC00        SUBROUTINE CORREC                                      *
*C                COMPUTE NON-FREQUENCY DEPENDENT CORRECTIONS            *
*      SUBROUTINE                                                        *
*     S           CORREC                                                 *
*     S           ( DUMMY1, K , M )                                      *
*C                                                                       *
*C     *******************************************************************
*C     *                                                                **
*C     CORRECTION FACTORS FOR RAY TYPES                                  *
*C     *                                                                **
*C     *******************************************************************
*C                                                                       *
*      COMMON / ANGRAD / ANGBER(2)                                       *
*      COMMON                                                            *
*     C / ARRAYC /                                                       *
*     D           NARRAY(2)                                              *
*     C           ARRYH1,                                                *
*     C           ARRYH2,                                                *
*     C           ARRYW1,                                                *
*     C           ARRYW2,                                                *
*     C           DELF   ,                                               *
*     C           FRES1 ,                                                *
*     C           FRES2 ,                                                *
*     C           QTRAN1,                                                *
*     C           QTRAN2,                                                *
*     NDUM2                                                              *
*      COMMON                                                            *
*     C / BEAMCR /                                                       *
*     D           BEMCOR(3,8)                                            *
*      COMMON                                                            *
*     C / ARRAYP /                                                       *
*     D           ANGDGA(2)        .                                     *
*     D           ARRAYD(3,2)      .                                     *
*     D           COSPHI(2)        .                                     *
*     D           COSRAD(2)        .                                     *
*     C           MSHIPS,                                                *
*     D           SINPHI(2)        .                                     *
*     D           SINRAD(2)        .                                     *
*     D           TSAR1 (3)        .                                     *
*     D           TSAR2 (3)        .                                     *
*     D           TMATRX(3,3,2)    .                                     *
*     D           TVECTR(3,2)      .                                     *
*     DDUM8                                                              *
*C                                                                       *
*      DIMENSION                                                         *
*     D           ANGSIG(3,2)      .                                     *
*     D           SAR1  (3)        .                                     *
*     D           SAR2  (3)        .                                     *
*     D           TARRIV(3,2)                                            *
*C                                                                       *
*      EQUIVALENCE                                                       *
*     Q ( TARRIV, TSAR1 ),                                               *
*     Q ( ANGSIG, SAR1 ),                                                *
*     Q ( ANGSIG(1,2),SAR2 )                                             *
*C                                                                       *
*C                                                                       *
*      C1 = COS( DUMMY1 )                                                *
*      ANGSIG(3,M) = SIN( DUMMY1 )                                       *
**************************************************************************
                                    I
                                    I
                                    I
                                    I
                                    I
**************************************************************************
*      IF( NARRAY(M) .EQ. 2 )                                            *......O
*      * GO TO 10                                                        *      I
**************************************************************************      I
                                    I                                           I
                                    I                                           I
                                    I                                           I
**************************************************************************      I
*      ANGSIG(2,M) = C1*SINRAD(M)                                        *      I
*      ANGSIG(1,M) = C1*COSRAD(M)                                        *      I
**************************************************************************      I
                                    I                                           I
                                    I                                           I
```



```
      DO 200
   I            I = 1,3

      TARRIV(I,M) = 0.0

      DO 100
   I             J = 1,3

      TARRIV(I,M) = TMATRX(I,J,M)*ANGSIG(J,M)
     E + TARRIV(I,M)

  100 CONTINUE

      BEMCOR(I,K) = ARRAYD(I,M)*( TARRIV(I,M) - TVECTR(I,M) )

  200 CONTINUE

      RETURN

C

   10 CONTINUE
      C2 = ANGDGA(M) - ANGBER(M)
      BEMCOR(1,K) = ARRAYD(1,M)*C1*COS( C2 )
      BEMCOR(2,K) = ARRAYD(2,M)*C1*SIN( C2 )
      BEMCOR(3,K) = ARRAYD(3,M)*ANGSIG(3,M)

      RETURN

C

      END
```

```
                                        (ENTRANCE)
                                            I
                                            I
*********************************************************************************
*CDEMUVA00         SUBROUTINE DEMUVA                                             *
*CDEMUV000         SUBROUTINE FOR COMPUTING THE MODIFIED PARAMETERS              *
*      SUBROUTINE                                                               *
*     S            DEMUVA                                                       *
*     S ( NDSTAT, NESTAT )                                                      *
*C                                                                              *
*C     ***************************************************************************
*C     *                                                                       **
*C     *                                                                       **
*C     *                                                                       **
*C     ***************************************************************************
*C                                                                              *
*      COMMON                                                                   *
*     C / SIGNAL /                                                              *
*     D            PRYSEV(128)      ,                                           *
*     D            PRYSSE(128)      ,                                           *
*     D            PRNOEV(128)      ,                                           *
*     D            PRNOSE(128)      ,                                           *
*     D            PROEVA(128)      ,                                           *
*     D            PROSER(128)      ,                                           *
*     D            VAREVA(128)      ,                                           *
*     D            VARSER(128)      ,                                           *
*     D            GMUEVA(128)      ,                                           *
*     D            GMUSER(128)      ,                                           *
*     D            DEVAEV(128)      ,                                           *
*     D            DEVASE(128)      ,                                           *
*     D            DEMUEV(128)      ,                                           *
*     D            DEMUSE(128)      ,                                           *
*     C            THREVA,                                                      *
*     C            THRSER,                                                      *
*     C            NTIMEN                                                       *
*C                                                                              *
*      EQUIVALENCE                                                              *
*     Q ( CTWOPI, Z1 )                                                          *
*C                                                                              *
*C     GMUEVA( )  = MEAN OF SMOOTHED EVADER S/N                                 *
*C     GMUSER( )  = MEAN OF SMOOTHED SEARCHER S/N                               *
*C     NDSTAT     = INTEGER VALUE OF THE D-STATE                                *
*C     NESTAT     = INTEGER VALUE OF THE E-STATE                                *
*C     VAREVA( )  = VARIANCE OF SMOOTHED S/N FOR EVADER                         *
*C     VARSER( )  = VARIANCE OF SMOOTHED S/N FOR SEARCHER                       *
*C     PROSER( )  = PROBABILITY OF DETECTION BY THE SEARCHER                    *
*C     PROEVA( )  = PROBABILITY OF DETECTION BY THE EVADER                      *
*C     THRSER     = DETECTION THRESHOLD FOR THE SEARCHER                        *
*C     THREVA     = DETECTION THRESHOLD FOR THE EVADER                          *
*C     PRNOSE( )  = PROBABILITY THAT DECISION IS NO DETECTION BY SEARCHER*
*C     PRNOEV( )  = PROBABILITY THAT DECISION IS NO DETECTION BY EVADER  *
*C     PRYSSE( )  = PROBABILITY THAT DECISION IS A DETECTION BY SEARCHER *
*C     PRYSEV( )  = PROBABILITY THAT DECISION IS A DETECTION BY EVADER   *
*C     DEMUSE( )  = MODIFIED MEAN OF S/N FOR THE SEARCHER                       *
*C     DEMUEV( )  = MODIFIED MEAN OF S/N FOR THE EVADER                         *
*C     DEVASE( )  = MODIFIED VARIANCE OF S/N FOR THE SEARCHER                   *
*C     DEVAEV( )  = MODIFIED VARIANCE OF S/N FOR THE SEARCHER                   *
*C                                                                              *
*      Z1 = 6.2831853                                                           *
*      DEMUEV(NTIMEN) = GMUEVA(NTIMEN)                                          *
*      Z3 = ( THREVA - GMUEVA(NTIMEN))**2/2.0/VAREVA(NTIMEN)                    *
*********************************************************************************
                                            I
                                            I
                                            I
                                            I
```

```
                                    I
*************************************************************************
*      IF( Z3 .GT. 88.0 )                                               *......0
*    * GO TO 10                                                         *      I
*************************************************************************      I
                                    I                                          I
                                    I                                          I
                                    I                                          I
*************************************************************************      I
*      Z4 = PROEVA(NTIMEN)                                              *      I
*      Z5 = 0.0                                                         *      I
*      IF( Z4 .NE. 0.0 ) Z5 = PRYSEV(NESTAT)/Z4                         *      I
*      IF( Z4 .NE. 1.0 )  Z5 = Z5 - PRNDEV(NESTAT)/( 1.0 - Z4 )         *      I
*      DEMUEV(NTIMEN)=DEMUEV(NTIMEN)+SQRT(VAREVA(NTIMEN)/Z1)*Z5/EXP(Z3) *      I
*************************************************************************      I
                                    I                                          I
                                    0(.........................................0
                                    I
*************************************************************************
*   10 CONTINUE                                                         *
*      DEMUSE(NTIMEN) = GMUSER(NTIMEN)                                  *
*      Z2 = ( THRSER - GMUSER(NTIMEN))**2/2.0/VARSER(NTIMEN)            *
*************************************************************************
                                    I
                                    I
                                    I
*************************************************************************
*      IF( Z2 .GT. 88.0 )                                               *......0
*    * GO TO 20                                                         *      I
*************************************************************************      I
                                    I                                          I
                                    I                                          I
                                    I                                          I
*************************************************************************      I
*      Z4 = PROSER(NTIMEN)                                              *      I
*      Z5 = 0.0                                                         *      I
*      IF( Z4 .NE. 0.0 ) Z5 = PRYSSE(NDSTAT)/Z4                         *      I
*      IF( Z4 .NE. 1.0 )  Z5 = Z5 - PRNDSE(NDSTAT)/( 1.0 - Z4 )         *      I
*      DEMUSE(NTIMEN)=DEMUSE(NTIMEN)+SQRT(VARSER(NTIMEN)/Z1)*Z5/EXP(Z2) *      I
*************************************************************************      I
                                    I                                          I
                                    0(.........................................0
                                    I
*************************************************************************
*   20 CONTINUE                                                         *
*      DEVASE(NTIMEN) = VARSER(NTIMEN) + ( THRSER - DEMUSE(NTIMEN) )*   *
*     E ( DEMUSE(NTIMEN) - GMUSER(NTIMEN) )                             *
*      DEVAEV(NTIMEN) = VAREVA(NTIMEN) + ( THREVA - DEMUEV(NTIMEN) )*   *
*     E ( DEMUEV(NTIMEN) - GMUEVA(NTIMEN) )                             *
*************************************************************************
                                    I
                                    I
                                    I
                                    I
                                    I
*************************************************************************
*      RETURN                                                           *
*************************************************************************


*************************************************************************
*C                                                                      *
*************************************************************************
                                    I
                                    I
                                    I
*************************************************************************
*      END                                                              *
*************************************************************************
```

```
                                   (ENTRANCE)
                                       I
                                       I
*************************************************************************
*CEXCHNG00        SUBROUTINE  EXCHNG                                    *
*CSEXCH000        SUBROUTINE FOR EXCHANGING TWO VALUE                   *
*     SUBROUTINE                                                        *
*    S            EXCHNG                                                *
*    S ( FIRSTO, SECOND )                                               *
*C                                                                      *
*C    *******************************************************************
*C    *                                                                **
*C    THIS EXCHANGES THE VALUES OF THE INPUTS                           *
*C    *                                                                **
*C    *******************************************************************
*C                                                                      *
*     DUMMY1 = FIRSTO                                                   *
*     FIRSTO = SECOND                                                   *
*     SECOND = DUMMY1                                                   *
*************************************************************************
                                       I
                                       I
                                       I
*************************************************************************
*     RETURN                                                            *
*************************************************************************


*************************************************************************
*C                                                                      *
*************************************************************************
                                       I
                                       I
                                       I
*************************************************************************
*     END                                                               *
*************************************************************************
```

```
                                  (ENTRANCE)
                                       I
                                       I
******************************************************************
*CFILT        PRE-DETECT. FILTER RESPONSE                        *
*C   *************************************************************
*C   *                                                           *
*C   *  SUB. FOR COMPUTING PRE-DETECT. FILTER RESPONSE           *
*C   *                                                           *
*C   *************************************************************
*       SUBROUTINE FILTER (FO,FBW,F,Y)                           *
*       Y=1./(1.+((F-FO)/FBW)**2)                                *
******************************************************************
                                       I
                                       I
                                       I
******************************************************************
*       RETURN                                                   *
******************************************************************


******************************************************************
*       END                                                      *
******************************************************************
```

```
                                   (ENTRANCE)
                                       I
                                       I
***********************************************************************
*CFVELOC00          FUNCTION   FVELOC                                 *
*CFVELOC000         FUNCTION FOR COMPUTING THE VELOCITY AT ANY DEPTH  *
*       FUNCTION   FVELOC( DELTAD, LAYERN )                           *
*C                                                                    *
*C      ***************************************************************
*C      *                                                            **
*C      THIS COMPUTES THE VELOCITY (IN K-YD/SEC) AT ANY POINT IN THE LAYER*
*C      *                                                            **
*C      ***************************************************************
*C                                                                    *
*       COMMON                                                        *
*      C / LCONST /                                                   *
*      C              NULAPO,                                         *
*      C              DEPROT,                                         *
*      D              CONSG0(128)       ,                             *
*      D              CONSG1(128)       ,                             *
*      D              CONSG2(128)       ,                             *
*      D              CONSV0(128)       ,                             *
*      D              DELTAZ(128)       ,                             *
*      D              DEPKYD(128)       ,                             *
*      D              SLOPEJ(128)       ,                             *
*      D              SPDKYD(128)                                     *
*C                                                                    *
*C      DELTAD      = DEPTH FROM TOP OF LAYER TO POINT OF INTEREST (IN K-YD*
*C      LAYERN      = NUMBER OF LAYER  IN WHICH VELOCITY IS SOUGHT (DIMENSI*
*C                                                                    *
*       FVELOC = 1.0/SQRT( CONSV0(LAYERN) + DELTAD*( CONSG0(LAYERN) +  *
*      E DELTAD*CONSG1(LAYERN) )/( 1.0 + CONSG2(LAYERN)*DELTAD )**2 )  *
***********************************************************************
                                       I
                                       I
                                       I
***********************************************************************
*       RETURN                                                        *
***********************************************************************


***********************************************************************
*C                                                                    *
***********************************************************************
                                       I
                                       I
                                       I
***********************************************************************
*       END                                                           *
***********************************************************************
```

```
                                     (ENTRANCE)
                                          I
                                          I
**********************************************************************
*CINIT        INITIAL VALUES                                         *
*C   ****************************************************************
*C   *                                                               *
*C   * TO INITIALIZE VALUES                                          *
*C   *                                                               *
*C   ****************************************************************
*       SUBROUTINE INIT                                              *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      44),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,SINPSE,SINPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,F0S,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                  *
*       COMMON / INTSPD / VXEINT, VYEINT                              *
*C                                                                   *
*C      VXEINT     = INITIAL EVADER SPEED IN X-DIRECTION            (K*
*C      VYEINT     = INITIAL EVADER SPEED IN Y-DIRECTION            (K*
*C                                                                   *
*       GAM=HE1-HS1                                                  *
*       SG=SIN(GAM)                                                  *
*       CG=COS(GAM)                                                  *
**********************************************************************
                                          I
                                          I
                                          I
**********************************************************************
*       IF (HE1.NE.HS1) GO TO 10                                     *......O
**********************************************************************      I
                                          I                                 I
                                          I                                 I
                                          I                                 I
**********************************************************************      I
*       PXE(1) = 0.                                                  *      I
*       PYE (1) = RI                                                 *      I
**********************************************************************      I
                                          I                                 I
                                          I                                 I
                                          I                                 I
**********************************************************************      I
*       GO TO 20                                                     *......I.......O
**********************************************************************      I       I
                                          O(................................O       I
                                          I                                         I
**********************************************************************              I
*    10 SS=SE1*SS1                                                   *              I
*       SSS=SS*SG                                                    *              I
*       SSC=SS*CG                                                    *              I
*       AS=ABS(SSS)                                                  *              I
*       SQ = SQRT(RI**2-RCJ**2)                                      *              I
*       SE2=SE1**2                                                   *              I
*       SS2=SS1**2                                                   *              I
*       AK=AS*SQRT(SE2+SS2-2.*SSC)                                   *              I
*       AY = RCJ*SS1*(SE2-SSC)+RCJ*SE1*CG*(SS2-SSC)-(AS*SQ)*(SE1*CG-SS1) *          I
*       AX = (SE1*SG)*(RCJ*(SS2-SSC)-AS*SQ)                          *              I
*       PYE(1)=AY/AK                                                 *              I
*       PXE(1)=AX/AK                                                 *              I
**********************************************************************              I
                                          I                                         I
                                          O(........................................O
                                          I
**********************************************************************
*    20 VXE(1)=SE1*SG                                                *
*       VYE(1)=SE1*CG                                                *
*       VXS(1)=0.                                                    *
*       VYS(1)=SS1                                                   *
*       PZE(1)=EDEPTH                                                *
*       PZS(1) = SDEPTH                                              *
*       PYS(1)=0.                                                    *
*       PXS(1)=0.                                                    *
*       VXEINT = VXE(1)                                              *
*       VYEINT = VYE(1)                                              *
*       ENTRY RESET                                                  *
*       CLPH = 0.0                                                   *
*       EVPH = 0.0                                                   *
**********************************************************************
                                          I
                                          I
                                          I
**********************************************************************
*       RETURN                                                       *
**********************************************************************


**********************************************************************
*       END                                                          *
**********************************************************************
```

```
                                      (ENTRANCE)
                                           I
                                           I
*************************************************************************
*CINTEG       SIGNA/NOISE RATIO                                         *
*C   ********************************************************************
*C   *                                                                  *
*C   *INTEGRATION FOR COMPUTING THE SIGNAL TO NOISE RATIO               *
*C   *                                                                  *
*C   ********************************************************************
*       SUBROUTINE PSIGP                                                *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128);P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,FOS,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBWE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                *
*       COMMON /TWOB2/ B2S,B2E                                          *
*C   ********************************************************************
*C   *                                                                  *
*C   *     FOS = CENTER FREQ. OF SEARCHER PREDETECT  FILTER             *
*C   *     F1S = LOWER LIMIT OF INTEG. FORSEARCHER                      *
*C   *     F2S = UPPER LIMIT OF INTEG. FORSEARCHER                      *
*C     FILEVA      = FUNCTION FOR FILTER RESPONSE OF EVADER             *
*C     FILSER      = FUNCTION FOR FILTER RESPONSE OF SEARCHER           *
*C   * PTS(K) = SIG. SPECTRUM SEARCHER  SEES                            *
*C   * FXS(K) =TABLE ORDERED  FROM LOW TO HIGH FREQ.                    *
*C   * PNS(K) =NOISE  SPECTRUM  SEARCHERSEES                            *
*C   * FNS(K) =TABLE ORDERED FROM LOW TOHIGH FREQ.                      *
*C   * FBWS = BANDWIDTH OF SEARCHER PRE-DETECT. FILTER                  *
*C     FOE =CENTER FREQ. OF EVADER  PRE-DETECT. FILTER                  *
*C   * FBWE =BANDWIDTH OF PREDETECT. FILTER                             *
*C   * F1E  =LOWER LIMIT OF INTEG. FOR  EVADER                          *
*C   * F2E = LOWER LIMIT OF INTEG. FOR  EVADER                          *
*C   * PTE(K) =SIG. SPECTRUM EVADER SEES                                *
*C   * FXE(K)=TABLE ORDERED FROM LOW TO HIGH FREQ.                      *
*C   * PNE(K) =NOISE SPECTRUM EVADER SEES                               *
*C   * FNE(K)= TABLE ORDERED FROM LOW TO HIGH FREQ.                     *
*C   *                                                                  *
*C   ********************************************************************
*       NAMELIST / RATIOS / AINPXS,AINPXE,AINPNS,AINPNE,XS,XE           *
*C                                                                      *
*       FILEVA( DUMMY1 ) = 1.0/( 1.0 + ( ( DUMMY1 - FOE )/FBWE )**2 )   *
*       FILSER( DUMMY1 ) = 1.0/( 1.0 + ( ( DUMMY1 - FOS )/FBWS )**2 )   *
*C                                                                      *
*       AINPXS=0.                                                       *
*       K=1                                                             *
*       FXAS=F1S                                                        *
*************************************************************************
                                           I
                                           O(............................O
                                           I                             I
                                           I                             I
                                           I                             I
                                           I                             I
*************************************************************************I
*   15 AFXS=FXS(K)                                                      *I
*************************************************************************I
                                           I                             I
                                           I                             I
                                           I                             I
*************************************************************************I
*       IF (AFXS.GT.F1S) GO TO 10                                       *.......I.......O
*************************************************************************       I       I
                                           I                             I       I
                                           I                             I       I
                                           I                             I       I
*************************************************************************I       I
*       K=K+1                                                           *I       I
*************************************************************************I       I
                                           I                             I       I
```

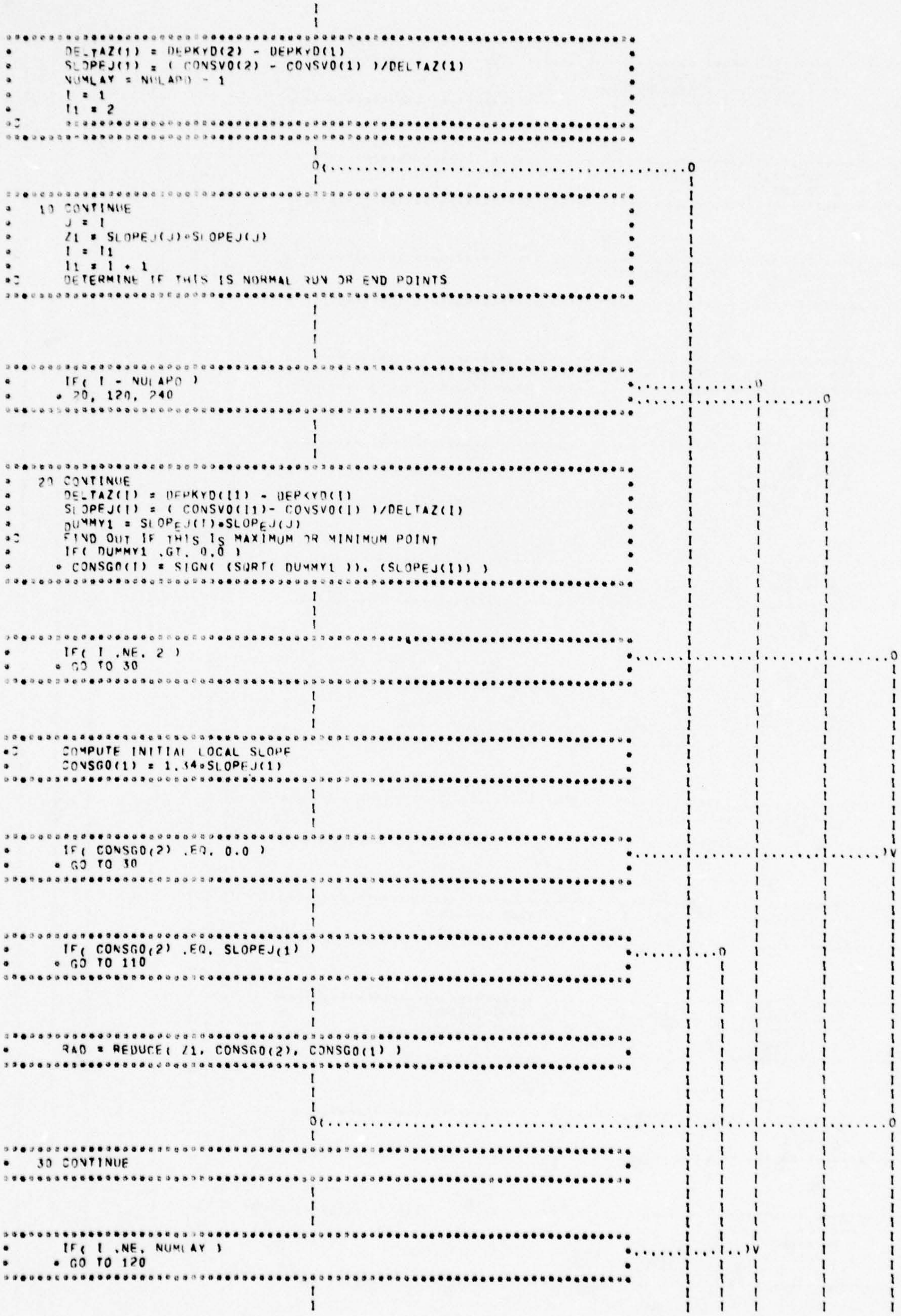
```
                                          I                                                 I        I
                                          I                                                 I        I
**************************************************************************************      I        I
*      GO TO 15                                                                      *.....)A        I
**************************************************************************************      I        I
                                                                                            I        I
                                          O(................................................I........O
                                          I                                                 I
**************************************************************************************      I
*   10 IF (AFXS.GT.F2S) GO TO 20                                                     *......I........O
**************************************************************************************      I        I
                                          I                                                 I        I
                                          I                                                 I        I
                                          I                                                 I        I
**************************************************************************************      I        I
*      CALL FILTER (F0S,FBWS,AFXS,YXS)                                               *      I        I
*      ARGPXS =PTS(K)*YXS                                                            *      I        I
*      AINPXS=AINPXS+ARGPXS*(AFXS-FXAS)                                              *      I        I
*      FXAS=AFXS                                                                     *      I        I
*      K=K+1                                                                         *      I        I
**************************************************************************************      I        I
                                          I                                                 I        I
                                          I                                                 I        I
                                          I                                                 I        I
**************************************************************************************      I        I
*      GO TO 15                                                                      *......O        I
**************************************************************************************               I
                                                                                                     I
                                          O(.........................................................O
                                          I
**************************************************************************************
*   20 AINPNS =0.                                                                    *
*      K=1                                                                           *
*      FNAS =F1S                                                                     *
**************************************************************************************
                                          I
                                          I
                                          I
                                          O(................................................O
                                          I                                                 I
**************************************************************************************      I
*   25 AFNS=FNS(K)                                                                   *      I
**************************************************************************************      I
                                          I                                                 I
                                          I                                                 I
                                          I                                                 I
**************************************************************************************      I
*      IF(AFNS.GT.F1S) GO TO 30                                                      *......I........O
**************************************************************************************      I        I
                                          I                                                 I        I
                                          I                                                 I        I
                                          I                                                 I        I
**************************************************************************************      I        I
*      K=K+1                                                                         *      I        I
**************************************************************************************      I        I
                                          I                                                 I        I
                                          I                                                 I        I
                                          I                                                 I        I
**************************************************************************************      I        I
*      GO TO 25                                                                      *.....)A        I
**************************************************************************************      I        I
                                                                                            I        I
                                          O(................................................I........O
                                          I                                                 I
**************************************************************************************      I
*   30 IF(AFNS.GT.F2S) GO TO 35                                                      *......I........O
**************************************************************************************      I        I
                                          I                                                 I        I
                                          I                                                 I        I
                                          I                                                 I        I
```

```
***********************************************************************      I      I
*       CALL FILTER(FOS,FBWS,AFNS,YNS)                                *      I      I
*       ARGPNS =(PNS(K)*YNS)**2                                       *      I      I
*       AINPNS = AINPNS+ARGPNS*(AFNS-FNAS)                            *      I      I
*       FNAS=AFNS                                                     *      I      I
*       K=K+1                                                         *      I      I
***********************************************************************      I      I
                                   I                                         I      I
                                   I                                         I      I
                                   I                                         I      I
***********************************************************************      I      I
*       GO TO 25                                                      *......O      I
***********************************************************************             I
                                                                                    I
                                   O(...............................................O
                                   I
***********************************************************************
*   35 CONTINUE                                                       *
*       XS = 1.0 + AINPXS/SQRT( 0.002*AINPNS*B2S )                    *
*   37 AINPXE=0.                                                      *
*       K=1                                                           *
*       FXAE=F1E                                                      *
***********************************************************************
                                   I
                                   I
                                   I
                                   O(.........................................O
                                   I                                         I
***********************************************************************      I
*   40 AFXE=FXE(K)                                                    *      I
***********************************************************************      I
                                   I                                         I
                                   I                                         I
                                   I                                         I
***********************************************************************      I
*       IF(AFXE.GT.F1E) GO TO 45                                      *......I......O
***********************************************************************      I      I
                                   I                                         I      I
                                   I                                         I      I
                                   I                                         I      I
***********************************************************************      I      I
*       K=K+1                                                         *      I      I
***********************************************************************      I      I
                                   I                                         I      I
                                   I                                         I      I
                                   I                                         I      I
***********************************************************************      I      I
*       GO TO 40                                                      *.....)A      I
***********************************************************************      I      I
                                                                             I      I
                                   O(........................................I......O
                                   I                                         I
***********************************************************************      I
*   45 IF(AFXE.GT.F2E) GO TO 50                                       *......I......O
***********************************************************************      I      I
                                   I                                         I      I
                                   I                                         I      I
                                   I                                         I      I
***********************************************************************      I      I
*       AINPXE = AINPXE + PTE(K)*FILEVA( AFXE )*( AFXE - FXAE )       *      I      I
*       FXAE =AFXE                                                    *      I      I
*       K=K+1                                                         *      I      I
***********************************************************************      I      I
                                   I                                         I      I
                                   I                                         I      I
                                   I                                         I      I
***********************************************************************      I      I
*       GO TO 40                                                      *......O      I
***********************************************************************             I
                                                                                    I
                                   O(...............................................O
                                   I
***********************************************************************
*   50 AINPNE = 0.                                                    *
*       K=1                                                           *
*       FNAE=F1E                                                      *
***********************************************************************
                                   I
                                   I
```

```
                                    I
                                    0(...........................................0
                                    I                                           I
************************************************************************        I
*   55 AFNE=FNE(K)                                                     *        I
************************************************************************        I
                                    I                                           I
                                    I                                           I
                                    I                                           I
************************************************************************        I
*       IF (AFNE.GT.F1E) GO TO 60                                      *........I.......0
************************************************************************        I       I
                                    I                                           I       I
                                    I                                           I       I
                                    I                                           I       I
************************************************************************        I       I
*       K=K+1                                                          *        I       I
************************************************************************        I       I
                                    I                                           I       I
                                    I                                           I       I
                                    I                                           I       I
************************************************************************        I       I
*       GO TO 55                                                       *.......)A       I
************************************************************************        I       I
                                                                                I       I
                                    0(..........................................I.......0
                                    I                                           I
************************************************************************        I
*   60 IF (AFNE.GT.F2E) GO TO 65                                       *........I.......0
************************************************************************        I       I
                                    I                                           I       I
                                    I                                           I       I
                                    I                                           I       I
************************************************************************        I       I
*       CALL FILTER (FOE,FBWE,AFNE,YNE)                                *        I       I
*       ARGPNE=(PNE(K)*YNE)**2                                         *        I       I
*       AINPNE=AINPNE+ARGPNE*(AFNE-FNAE)                               *        I       I
*       FNAE=AFNE                                                      *        I       I
*       K=K+1                                                          *        I       I
************************************************************************        I       I
                                    I                                           I       I
                                    I                                           I       I
                                    I                                           I       I
************************************************************************        I       I
*       GO TO 55                                                       *........0       I
************************************************************************                I
                                                                                        I
                                    0(..................................................0
                                    I
************************************************************************
*   65 CONTINUE                                                        *
*       XE = 1,0 + AINPXE/SQRT( 0.002*AINPNE*B2E )                     *
************************************************************************
                                    I
                                    I
                                    I
                                    I
                                    I
************************************************************************
*       RETURN                                                         *
************************************************************************


************************************************************************
*       END                                                            *
************************************************************************
```

```
                                    (ENTRANCE)
                                        I
                                        I
*****************************************************************
*CINVB        INVERSE BEARING RIDER                             *
*       SUBROUTINE INVBRI                                       *
*       COMMON /LAREL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,SINPSE,SINPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,F0S,FBWS,F1S,PIS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*      7,F2S,PIE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PREP*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                        *
*       VXE(N-1)=SE1*ALPXN                                      *
*       VYE(N-1)=SE1*ALPYN                                      *
*****************************************************************
                                        I
                                        I
                                        I
*****************************************************************
*       RETURN                                                  *
*****************************************************************


*****************************************************************
*       END                                                     *
*****************************************************************
```

```
                                             (ENTRANCE)
                                                 I
                                                 I
                      *************************************************************************
                      *CLAYERS00          SUBROUTINE LAYERS                                   *
                      *CS.AYRC00          SUBROUTINE FOR LAYER CONSTANTS                      *
                      *      SUBROUTINE                                                       *
                      *     S             LAYERS                                              *
                      *     S             ( MAXLAY )                                          *
                      *C                                                                      *
                      *C     *****************************************************************
                      *C     *                                                               **
                      *C     THIS COMPUTES THE LAYER CONSTANTS                                *
                      *C     *                                                               **
                      *C     *****************************************************************
                      *C                                                                      *
                      *      COMMON                                                           *
                      *     C / LCONST /                                                      *
                      *     C             NULAPO,                                             *
                      *     C             DEPBOT,                                             *
                      *     D             CONSG0(128)       .                                 *
                      *     D             CONSG1(128)       .                                 *
                      *     D             CONSG2(128)       .                                 *
                      *     D             CONSV0(128)       .                                 *
                      *     D             DELTAZ(128)       .                                 *
                      *     D             DEPKYD(128)       .                                 *
                      *     D             SLOPEJ(128)       .                                 *
                      *     D             SPDKYD(128)                                         *
                      *C                                                                      *
                      *C     CONSG0( )  = G0 CONSTANT AS COMPUTED BY LAYERS ROUTINE    (SEC**2/*
                      *C     CONSG1( )  = G1 CONSTANT AS COMPUTED BY LAYERS ROUTINE    (SEC**2/*
                      *C     CONSG2( )  = G2 CONSTANT AS COMPUTED BY LAYERS ROUTINE         (K*
                      *C     CONSV0( )  = V0 CONSTANT AS COMPUTED BY LAYERS ROUTINE     ((SEC/K*
                      *C     DELTAZ( )  = DEPTH (IN K-YD) OF LAYER                            *
                      *C     DEPKYD( )  = DEPTH (IN K-YD) TO TOP OF LAYER FROM OCEAN SURFACE  *
                      *C     JUMPIF     = CONTROL PARAMETER FOR RECALCULATING A LAYER CONSTANT *
                      *C     MAXLAY     = MAXIMUM NUMBER OF LAYERS ALLOWED INCLUDING ADDED POIN*
                      *C     NSTART     = CONTROL PARAMETER FOR START OF LOOP                 *
                      *C     NULAPO     = NUMBER OF LAYERS PLUS ONE                           *
                      *C     NUMLAY     = NUMBER OF LAYERS                                    *
                      *C     RAD        =                                                     *
                      *C     SPDKYD( )  = SPEED OF SOUND PROPAGATION (IN K-YD/SEC)            *
                      *C     SLOPEJ( )  = SLOPE IN LAYER (IN SEC**2/K-YD**3 )                 *
                      *C     SQR        =                                                     *
                      *C     XI         =                                                     *
                      *C                                                                      *
                      *************************************************************************
                                                 I
                                                 I
                                                 I
                      *************************************************************************
I.....................*      DO 100                                                           *
I                     *     I             I = 1,NULAPO                                        *
I                     *************************************************************************
I                                                I
I                                                I
I                                                I
I                                                I
I                     *************************************************************************
I                     *      CONSV0(I) = 1.0/SPDKYD(I)/SPDKYD(I)                              *
I                     *      CONSG0(I) = 0.0                                                  *
I                     *************************************************************************
I                                                I
I                                                I
I                                                I
I                     *************************************************************************
......................*  100 CONTINUE                                                         *
                      *************************************************************************
                                                 I
                                                 I
                                                 I
                      *************************************************************************
                      *      DEPBOT = DEPKYD(NULAPO)                                          *
                      *      MAXLAY = MAX0( NULAPO + 10, MAXLAY )                             *
                      *************************************************************************
                                                 I
                                                 I
                                                 I
                      *************************************************************************
I.....................*      DO 200                                                           *
I                     *     I             I = NULAPO,MAXLAY                                   *
I                     *************************************************************************
I                                                I
I                                                I
I                                                I
I                     *************************************************************************
I                     *      CONSG0(I) = 0.0                                                  *
I                     *************************************************************************
I                                                I
I                                                I
I                                                I
I                     *************************************************************************
......................*  200 CONTINUE                                                         *
                      *************************************************************************
                                                 I
```



```
      DELTAZ(1) = DEPKYD(2) - DEPKYD(1)
      SLOPEJ(1) = ( CONSVO(2) - CONSVO(1) )/DELTAZ(1)
      NUMLAY = NULAPO - 1
      I = 1
      I1 = 2
C

   10 CONTINUE
      J = 1
      Z1 = SLOPEJ(J)*SLOPEJ(J)
      I = I1
      I1 = I + 1
C     DETERMINE IF THIS IS NORMAL RUN OR END POINTS

      IF( I - NULAPO )
     * 20, 120, 240

   20 CONTINUE
      DELTAZ(I) = DEPKYD(I1) - DEPKYD(I)
      SLOPEJ(I) = ( CONSVO(I1)- CONSVO(I) )/DELTAZ(I)
      DUMMY1 = SLOPEJ(I)*SLOPEJ(J)
C     FIND OUT IF THIS IS MAXIMUM OR MINIMUM POINT
      IF( DUMMY1 .GT. 0.0 )
     * CONSGO(I) = SIGN( (SQRT( DUMMY1 )), (SLOPEJ(I)) )

      IF( I .NE. 2 )
     * GO TO 30

C     COMPUTE INITIAL LOCAL SLOPE
      CONSGO(1) = 1.34*SLOPEJ(1)

      IF( CONSGO(2) .EQ. 0.0 )
     * GO TO 30

      IF( CONSGO(2) .EQ. SLOPEJ(1) )
     * GO TO 110

      RAD = REDUCE( Z1, CONSGO(2), CONSGO(1) )

   30 CONTINUE

      IF( I .NE. NUMLAY )
     * GO TO 120
```

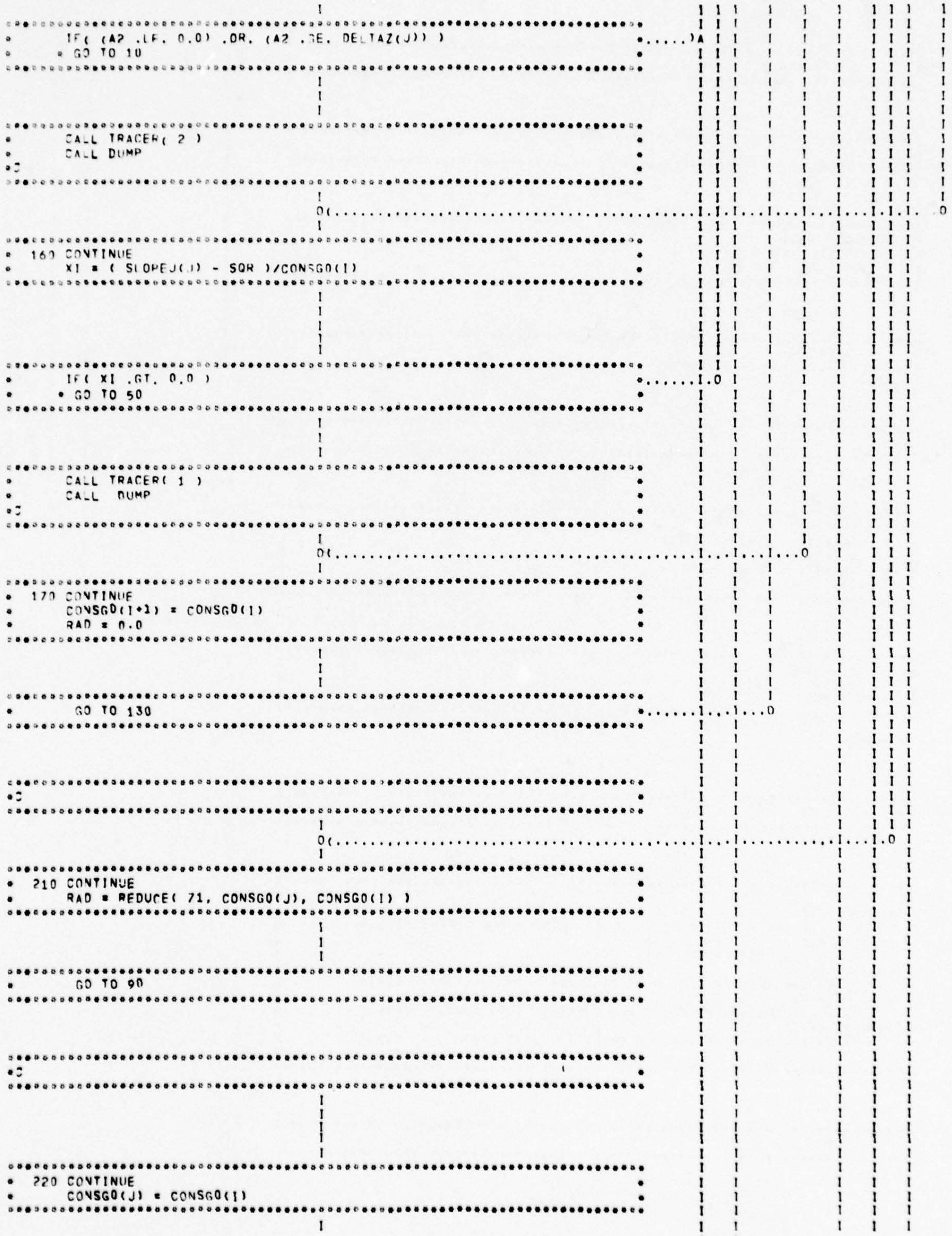
```
                                     I                                           I   I   I       I       I
***************************************************************************      I   I   I       I       I
*C     COMPUTE FINAL LOCAL SLOPE                                          *      I   I   I       I       I
*      CONSG0(NULAPO) = 1.34*SLOPEJ(I)                                    *      I   I   I       I       I
***************************************************************************      I   I   I       I       I
                                     I                                           I   I   I       I       I
                                     I                                           I   I   I       I       I
                                     I                                           I   I   I       I       I
***************************************************************************      I   I   I       I       I
*      IF( CONSG0(I) .EQ. 0.0 )                                           *......I...I..)V       I       I
*     * GO TO 120                                                         *      I   I   I       I       I
***************************************************************************      I   I   I       I       I
                                     I                                           I   I   I       I       I
                                     I                                           I   I   I       I       I
                                     I                                           I   I   I       I       I
***************************************************************************      I   I   I       I       I
*      IF( CONSG0(I) .EQ. SLOPEJ(I) )                                     *......I...I...I...0   I       I
*     * GO TO 170                                                         *      I   I   I   I   I       I
***************************************************************************      I   I   I   I   I       I
                                     I                                           I   I   I   I   I       I
                                     I                                           I   I   I   I   I       I
                                     I                                           I   I   I   I   I       I
***************************************************************************      I   I   I   I   I       I
*      RAD = REDUCE( SLOPEJ(I)**2, CONSG0(I), CONSG0(I1) )                *      I   I   I   I   I       I
***************************************************************************      I   I   I   I   I       I
                                     I                                           I   I   I   I   I       I
                                     0(..........................................I...I...I...I...I...0   I
                                     I                                           I   I   I   I   I   I   I
***************************************************************************      I   I   I   I   I   I   I
*   40 CONTINUE                                                           *      I   I   I   I   I   I   I
*      SQR = SQRT( RAD )                                                  *      I   I   I   I   I   I   I
*      XI = ( SLOPEJ(J) + SQR )/CONSG0(I)                                 *      I   I   I   I   I   I   I
***************************************************************************      I   I   I   I   I   I   I
                                     I                                           I   I   I   I   I   I   I
                                     I                                           I   I   I   I   I   I   I
                                     I                                           I   I   I   I   I   I   I
***************************************************************************      I   I   I   I   I   I   I
*      IF( XI .LE. 0.0 )                                                  *......I...I...I...I...I...I...I...0
*     * GO TO 160                                                         *      I   I   I   I   I   I   I   I
***************************************************************************      I   I   I   I   I   I   I   I
                                     I                                           I   I   I   I   I   I   I   I
                                     I                                           I   I   I   I   I   I   I   I
                                     I                                           I   I   I   I   I   I   I   I
                                     0(..........................................I.0 I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
*   50 CONTINUE                                                           *      I I I   I   I   I   I   I   I
*      CONSG2(J) = ( XI - 1.0 )/DELTAZ(J)                                 *      I I I   I   I   I   I   I   I
*      CONSG1(J) = ( SLOPEJ(J)*XI*XI - CONSG0(J) )/DELTAZ(J)              *      I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
*      IF( CONSG2(J) .EQ. 0.0 )                                           *.....)A I I   I   I   I   I   I   I
*     * GO TO 10                                                          *      I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
*      IF( (CONSG0(J) .EQ. 0.0) .AND. (CONSG1(J) .EQ. 0.0 ) )             *.....)A I I   I   I   I   I   I   I
*     * GO TO 10                                                          *      I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
*C     FIND POINT OF INFLECTION AND MAKE PROPER ADJUSTMENTS               *      I I I   I   I   I   I   I   I
*      A2 = ( CONSG1(J) - 2.0*CONSG0(J)*CONSG2(J) )/CONSG2(J)/            *      I I I   I   I   I   I   I   I
*     E ( 2.0*CONSG1(J) - CONSG0(J)*CONSG2(J) )                           *      I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
*      IF( (A2 .LE. 0.0) .OR. (A2 .GE. DELTAZ(J)) )                       *.....)A I I   I   I   I   I   I   I
*     * GO TO 10                                                          *      I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
*      WRITE( 6, 9000 )                                                   *      I I I   I   I   I   I   I   I
*     W DEPKYD(J),                                                        *      I I I   I   I   I   I   I   I
*     W DEPKYD(I)                                                         *      I I I   I   I   I   I   I   I
*      A2 = ( SLOPEJ(J) - SQR )/CONSG0(I)                                 *      I I I   I   I   I   I   I   I
***************************************************************************      I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
                                     I                                           I I I   I   I   I   I   I   I
```

```
                                       I
.....................................................................
.      IF( A2 .EQ. XI )                                             .......I.I.I...I.0
.      * GO TO 60                                                   .
.....................................................................
                                       I
                                       I
.....................................................................
.      XI = A2                                                      .
.....................................................................
                                       I
                                       I
.....................................................................
.      IF( XI .GT. 0.0 )                                            .......I)A
.      * GO TO 50                                                   .
.....................................................................
                                       I
                                       0(..................................I.I.I...I.0
                                       I
.....................................................................
.   60 CONTINUE                                                     .
.....................................................................
                                       I
                                       I
.....................................................................
.      IF( NULAPO .GE. MAXLAY )                                     .......I.I.I...I.I.I..)V
.      * GO TO 240                                                  .
.....................................................................
                                       I
                                       I
.....................................................................
.      DELTAZ(J) = DELTAZ(J)/2.0                                    .
.      DELTAZ(I) = DELTAZ(J)                                        .
.      Y1 = ( DEPKYD(I) + DEPKYD(J) )/2.0                           .
.      U1 = ( CONSVO(I) + CONSVO(J) )/2.0                           .
.....................................................................
                                       I
                                       0(..................................I.I.I...I.I.I...I.0
                                       I
.....................................................................
.   70 CONTINUE                                                     .
.C     THIS SECTION ADDS A NEW POINT                                .
.      WRITE ( 6, 6000 )                                            .
.     W DEPKYD(J),                                                  .
.     W DEPKYD(I)                                                   .
.      NUMLAY = NULAPO                                              .
.      NULAPO = NULAPO + 1                                          .
.      K = NULAPO - 1                                               .
.....................................................................
                                       I
                                       I
.....................................................................
I...................   DO 300                                       .
I     .      I             L = 1,K                                  .
I     ...............................................................
I                                      I
I                                      I
I     ...............................................................
I     .      J1 = NULAPO - L                                        .
I     .      J2 = J1 + 1                                            .
I     .      DEPKYD(J2) = DEPKYD(J1)                                .
I     .      CONSVO(J2) = CONSVO(J1)                                .
I     .      SPDKYD(J2) = SPDKYD(J1)                                .
I     ...............................................................
I                                      I
I     ...............................................................
I...................   300 CONTINUE                                 .
      ...............................................................
                                       I
                                       I
.....................................................................
.      DEPKYD(I) = Y1                                               .
.      CONSVO(I) = U1                                               .
.      SPDKYD(I) = 1.0/SQRT( U1 )                                   .
.....................................................................
                                       I
                                       I
.....................................................................
.      IF( J .EQ. I )                                               .......I.I.I...I.I.I...I.I.I...I.0
.      * GO TO 250                                                  .
.....................................................................
                                       I
                                       I
.....................................................................
.      SLOPEJ(I) = SLOPEJ(J)                                        .
.      A1 = CONSGO(I)                                               .
.....................................................................
                                       I
```

```
                                  I                                             I I I   I I I   I I I   I I I
                                  I                                             I I I   I I I   I I I   I I I
...................................................................            I I I   I I I   I I I   I I I
.      IF( A1 .EQ. 0.0 )                                           ........I.I.I.0 I I I   I I I   I I I
.      * GO TO 270                                                 .       I I I I I I I   I I I   I I I
...................................................................        I I I I I I I   I I I   I I I
                                  I                                        I I I I I I I   I I I   I I I
                                  I                                        I I I I I I I   I I I   I I I
                                  I                                        I I I I I I I   I I I   I I I
...................................................................        I I I I I I I   I I I   I I I
.      RAD = REDUCE( Z1, A1, CONSGO(I) )                           .       I I I I I I I   I I I   I I I
...................................................................        I I I I I I I   I I I   I I I
                                  I                                        I I I I I I I   I I I   I I I
                                  O(.......................................I.I.I.O I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
...................................................................        I I I   I I I   I I I   I I I
.  270 CONTINUE                                                    .       I I I   I I I   I I I   I I I
...................................................................        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
...................................................................        I I I   I I I   I I I   I I I
.      IF( CONSGO(J) .EQ. 0.0 )                                    ........I.I.I...I.I.I...I.I.I)A   I I I
.      * GO TO 40                                                  .       I I I   I I I   I I I   I I I
...................................................................        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
...................................................................        I I I   I I I   I I I   I I I
.      RAD = REDUCE( Z1, CONSGO(J), CONSGO(I) )                    .       I I I   I I I   I I I   I I I
...................................................................        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
...................................................................        I I I   I I I   I I I   I I I
.       GO TO 40                                                   ........I.I.I...I.I.I...I.I.I)A   I I I
...................................................................        I I I   I I I   I I I   I I I
                                  I                                        I I I   I I I   I I I   I I I
                                  O(.......................................I.I.I.O I I I   I I I   I I I
                                  I                                        I I I I I I I   I I I   I I I
...................................................................        I I I I I I I   I I I   I I I
.    80 CONTINUE                                                   .       I I I I I I I   I I I   I I I
.      CALL TRACER( 8 )                                            .       I I I I I I I   I I I   I I I
.      A1 = SLOPEJ(J)/CONSGO(I)                                    .       I I I I I I I   I I I   I I I
...................................................................        I I I I I I I   I I I   I I I
                                  I                                        I I I I I I I   I I I   I I I
                                  I                                        I I I I I I I   I I I   I I I
                                  I                                        I I I I I I I   I I I   I I I
...................................................................        I I I I I I I   I I I   I I I
.      IF( A1 .GT. 1.0 )                                           ........I.I.I.I.I.I.I.O I I I   I I I
.     X GO TO 180                                                  .       I I I I I I I I I I I   I I I
...................................................................        I I I I I I I I I I I   I I I
                                  I                                        I I I I I I I I I I I   I I I
                                  I                                        I I I I I I I I I I I   I I I
                                  I                                        I I I I I I I I I I I   I I I
...................................................................        I I I I I I I I I I I   I I I
.      A2 = SLOPEJ(J)/CONSGO(J)                                    .       I I I I I I I I I I I   I I I
...................................................................        I I I I I I I I I I I   I I I
                                  I                                        I I I I I I I I I I I   I I I
                                  I                                        I I I I I I I I I I I   I I I
                                  I                                        I I I I I I I I I I I   I I I
...................................................................        I I I I I I I I I I I   I I I
.      IF( ABS( 1.0 - A2 ) .GT. 1.0E-7 )                           ........I.I.I.I.I.I.I.I.I.I.I.O I I I
.      * GO TO 210                                                 .       I I I I I I I I I I I I I I I
...................................................................        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
...................................................................        I I I I I I I I I I I I I I I

TOO MANY EXITS IN GO TO STATEMENT
.      IF( A1 .EQ. 1.0 )                                           ........I.I.I.I.I.I.I.I.I.I.I.I)V I I I
.      * GO TO 220                                                 .       I I I I I I I I I I I I I I I
...................................................................        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
...................................................................        I I I I I I I I I I I I I I I
.      IF( A2 .EQ. 1.0 )                                           ........I.I.I.I.I)A I I I I I I I I I I
.      * GO TO 60                                                  .       I I I I I I I I I I I I I I I
...................................................................        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
...................................................................        I I I I I I I I I I I I I I I
.      CONSGO(I) = CONSGO(J)                                       .       I I I I I I I I I I I I I I I
...................................................................        I I I I I I I I I I I I I I I
                                  I                                        I I I I I I I I I I I I I I I
```

```
TOO MANY EXITS IN GO TO STATEMENT
                                                    O(...........................................
                                                    I
*****************************************************************************
*   90 CONTINUE                                                             *
*       RAD = Z1 - CONSGO(J)*CONSGO(I)                                      *
*****************************************************************************
                                                    I
                                                    I
                                                    I
*****************************************************************************
*       IF( RAD .GE. 0.0 )                                                  *.....A
*     * GO TO 40                                                            *
*****************************************************************************
                                                    I
                                                    I
                                                    I
*****************************************************************************
*       CALL TRACER( 4 )                                                    *
*       CALL DUMP                                                           *
*C                                                                          *
*****************************************************************************
                                                    I
                                                    O(...........................................O
                                                    I
*****************************************************************************
*  110 CONTINUE                                                             *
*       CONSGO(1) = CONSGO(2)                                               *
*****************************************************************************
                                                    I
                                                    I
                                                    I
*****************************************************************************
*        GO TO 30                                                           *.....O
*****************************************************************************



*****************************************************************************
*C                                                                          *
*****************************************************************************
                                                    I
                                                    O(...........................................O
                                                    I
*****************************************************************************
*  180 CONTINUE                                                             *
*       RAD = REDUCE( Z1, CONSGO(I), CONSGO(J) )                            *
*****************************************************************************
                                                    I
                                                    O(...........................................O
                                                    I
*****************************************************************************
*  190 CONTINUE                                                             *
*       I1 = I                                                              *
*       I = J                                                               *
*       J = I - 1                                                           *
*       Z1 = SLOPEJ(J)*SLOPEJ(J)                                            *
*****************************************************************************
                                                    I
                                                    O(...........................................O
                                                    I
*****************************************************************************
*  120 CONTINUE                                                             *
*       RAD = Z1 - CONSGO(J)*CONSGO(I)                                      *
*****************************************************************************
                                                    I
                                                    I
                                                    I
*****************************************************************************
*       IF( RAD .LT. 0.0 )                                                  *.....O
*     * GO TO 80                                                            *
*****************************************************************************
                                                    I
                                                    O(...........................................O
                                                    I
*****************************************************************************
*  130 CONTINUE                                                             *
*****************************************************************************
                                                    I
                                                    I
                                                    I
*****************************************************************************
*       IF( CONSGO(I) .NE. 0.0 )                                            *.....O
*     * GO TO 40                                                            *
*****************************************************************************
                                                    I
                                                    I
                                                    I
*****************************************************************************
*  140 CONTINUE                                                             *
*       WRITE ( 6, 10000 )                                                  *
*     W DEPKYD(I)                                                           *
*****************************************************************************
                                                    I
```

```
                                  I
                                  I
                                  I
***************************************************************
*      IF( SLOPEJ(J) .EQ. 0.0 )                               *......I.I.O
*      * GO TO 230                                            *
***************************************************************
                                  I
                                  I
                                  I
***************************************************************
*      A2 = SLOPEJ(J)/CONSG0(J)                               *
***************************************************************
                                  I
                                  I
                                  I
***************************************************************
*      IF( A2 .GT. 1.0 )                                      *......I.I.I...I.O
*      * GO TO 60                                             *
***************************************************************
                                  I
                                  I
                                  I
***************************************************************
*      IF( A2 .EQ. 1.0 )                                      *......I.I.I...I...I...I.I.O
*      * GO TO 260                                            *
***************************************************************
                                  I
                                  I
                                  I
***************************************************************
*      A1 = 1.0 - 0.25/( A2 - 0.5 )                           *
***************************************************************
                                  I
                                  I
                                  I
***************************************************************
*      IF( (A1 .LE. 0.0) .OR. (A1 .GE. 1.0) )                 *......I.I.I...I.O
*      * GO TO 150                                            *
***************************************************************
                                  I
                                  I
                                  I
***************************************************************
*      WRITE( 6, 9000 )                                       *
*     W DEPKYD(J),                                            *
*     W DEPKYD(I)                                             *
*      A1 = CONSV0(I) - CONSV0(J)                             *
*      DZNEW = 2.5*A1/CONSG0(J) - 1.5*DELTAZ(J)               *
*      U1 = CONSV0(J) + CONSG0(J)*DZNEW                       *
*      Y1 = DEPKYD(J) + DZNEW                                 *
*      CONSG0(I) = CONSG0(J)                                  *
*      CONSG1(J) = 0.0                                        *
*      CONSG2(J) = 0.0                                        *
*      SLOPEJ(I1) = SLOPEJ(I)                                 *
*      SLOPEJ(J) = CONSG0(J)                                  *
*      DELTAZ(I) = DELTAZ(J) - DZNEW                          *
*      DELTAZ(J) = DZNEW                                      *
*      J = I                                                  *
***************************************************************
                                  I
                                  I
                                  I
                                  I
                                  I
***************************************************************
*      GO TO 70                                               *......I.I.I...I.I.I...I.O
***************************************************************

                                  O(..............................I.I.I...I.I.I...I...I.I.O
                                  I
***************************************************************
*  250 CONTINUE                                               *
*      SLOPEJ(I) = ( CONSV0(I1) - CONSV0(I) )/DELTAZ(I)       *
*      I = I1                                                 *
*      I1 = I+ 1                                              *
*      CONSG0(I) = 0.0                                        *
*      DELTAZ(I) = DEPKYD(I1) - DEPKYD(I)                     *
***************************************************************
                                  I
                                  O(..............................I.I.I...I.O
                                  I
***************************************************************
*  150 CONTINUE                                               *
*      XI = CONSG0(J)/2.0/SLOPEJ(J)                           *
*      CONSG1(J) = ( SLOPEJ(J)*XI*XI - CONSG0(J) )/DELTAZ(J)  *
*      CONSG2(J) = ( XI - 1.0 )/DELTAZ(J)                     *
*      A2 = CONSG0(J)*CONSG2(J)                               *
*      A3 = CONSG1(J) - A2                                    *
*      A2 = ( A3 - A2 )/CONSG2(J)/( CONSG1(J) + A3 )          *
***************************************************************
                                  I
                                  I
```



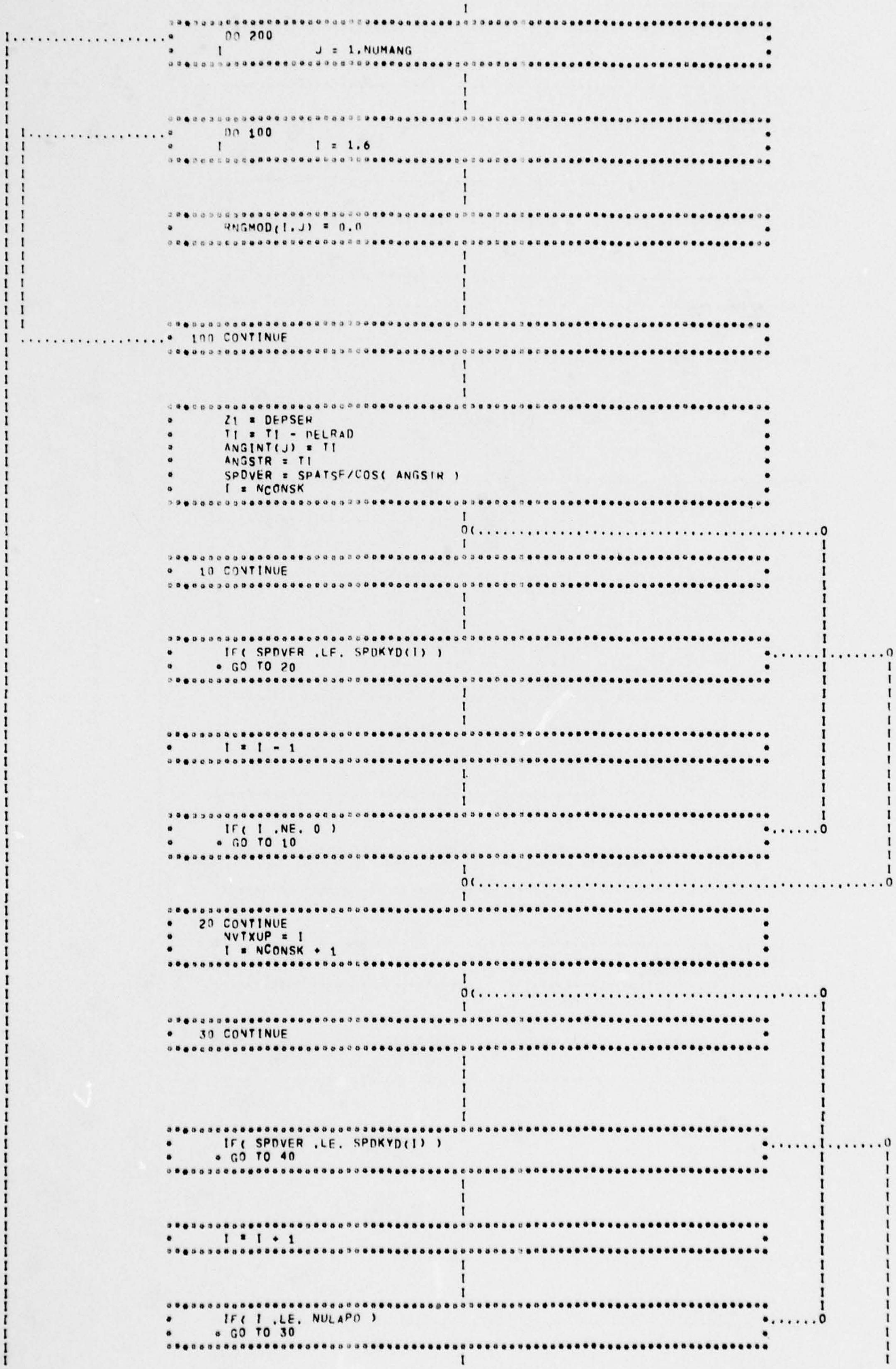
```
      IF( (A2 .LF. 0.0) .OR. (A2 .GE. DELTAZ(J)) )
     * GO TO 10

      CALL TRACER( 2 )
      CALL DUMP
C

  160 CONTINUE
      XI = ( SLOPEJ(J) - SQR )/CONSGO(I)

      IF( XI .GT. 0.0 )
     * GO TO 50

      CALL TRACER( 1 )
      CALL  DUMP
C

  170 CONTINUE
      CONSGO(I+1) = CONSGO(I)
      RAD = 0.0

       GO TO 130

C

  210 CONTINUE
      RAD = REDUCE( 71, CONSGO(J), CONSGO(I) )

       GO TO 90

C

  220 CONTINUE
      CONSGO(J) = CONSGO(I)
```

```
                                  I                                       I   I          I   I   I
                                  I                                       I   I          I   I   I
*********************************************************************     I   I          I   I   I
*       GO TO 190                                                   *.....I...I..........I...I...O
*********************************************************************     I   I          I   I
                                                                          I   I          I   I
                                                                          I   I          I   I
                                                                          I   I          I   I
*********************************************************************     I   I          I   I
*C                                                                  *     I   I          I   I
*********************************************************************     I   I          I   I
                                  I                                       I   I          I   I
                                  O(......................................I...O          I   I
                                  I                                       I              I   I
*********************************************************************     I              I   I
*  230 CONTINUE                                                     *     I              I   I
*      CONSG1(J) = 0.0                                              *     I              I   I
*      CONSG2(J) = 0.0                                              *     I              I   I
*********************************************************************     I              I   I
                                  I                                       I              I   I
                                  I                                       I              I   I
                                  I                                       I              I   I
*********************************************************************     I              I   I
*       GO TO 10                                                    *.....O              I   I
*********************************************************************                    I   I
                                                                                         I   I
                                                                                         I   I
                                                                                         I   I
                                                                                         I   I
*********************************************************************                    I   I
*C                                                                  *                    I   I
*********************************************************************                    I   I
                                  I                                                      I   I
                                  O(.....................................................I...O
                                  I                                                      I
*********************************************************************                    I
*  260 CONTINUE                                                     *                    I
*      I1 = I - NULAPO + MAXLAY - 11                                *                    I
*      NULAPO = MAXLAY                                              *                    I
*      WRITE ( 6, 7000 )                                            *                    I
*     W I1                                                          *                    I
*********************************************************************                    I
                                  I                                                      I
                                  O(.....................................................O
                                  I
*********************************************************************
*  240 CONTINUE                                                     *
*      WRITE ( 6, 8000 )                                            *
*     W DEPBOT                                                      *
*********************************************************************
                                  I
                                  I
                                  I
                                  I
                                  I
*********************************************************************
*       RETURN                                                      *
*********************************************************************


*********************************************************************
*C                                                                  *
* 6000 FORMAT(                                                      *
*      F/46H A NEW POINT IS BEING ADDED BETWEEN THE DEPTHS 2E15.6)  *
* 7000 FORMAT(                                                      *
*      F 22H NEW POINT NEEDED AT     I4 )                           *
* 8000 FORMAT(                                                      *
*      F/46H THE CONSTANT MAXIMUM OCEAN DEPTH IS AT (K-YD)  E15.6)  *
* 9000 FORMAT(                                                      *
*      F/46H AN INFLECTION POINT WAS FOUND BETWEEN POINTS  2E15.6)  *
*10000 FORMAT(                                                      *
*      F/46H A MAXIMUM OR MINIMUM POINT IS LOCATED AT       E15.6)  *
*C                                                                  *
*********************************************************************
                                  I
                                  I
                                  I
*********************************************************************
*       END                                                         *
*********************************************************************
```

```
                                    (ENTRANCE)
                                         I
                                         I
*************************************************************************
*CLENGTH00         SUBROUTINE LENGTH                                    *
*C                 FOR COMPUTING PATHLENGTHS, TIME AND DERIVATIVES      *
*       SUBROUTINE                                                      *
*      S           LENGTH                                               *
*      S ( N, M, DUMMY1, DUMMY2 )                                       *
*C                                                                      *
*C      *****************************************************************
*C      *                                                              **
*C      *                                                              **
*C      *****************************************************************
*C                                                                      *
*       COMMON                                                          *
*      C / CONSTN /                                                     *
*      C           DEPSER,                                              *
*      C           NCONSK,                                              *
*      C           SPATSF,                                              *
*      C           DEPEVA,                                              *
*      C           NCONSL,                                              *
*      C           SPATEV                                               *
*       COMMON                                                          *
*      C / LCONST /                                                     *
*      C           NULAPO,                                              *
*      C           DEPROT,                                              *
*      D           CONSG0(128)      ,                                   *
*      D           CONSG1(128)      ,                                   *
*      D           CONSG2(128)      ,                                   *
*      D           CONSV0(128)      ,                                   *
*      D           DELTAZ(128)      ,                                   *
*      D           DEPKYD(128)      ,                                   *
*      D           SLOPEJ(128)      ,                                   *
*      D           SPDKYD(128)                                          *
*       COMMON                                                          *
*      C / PTHLNG /                                                     *
*      C           A1     ,                                             *
*      C           BI     ,                                             *
*      C           CDSORD,                                              *
*      C           CI     ,                                             *
*      C           DI     ,                                             *
*      C           DXDC   ,                                             *
*      C           DZ1    ,                                             *
*      C           DZM    ,                                             *
*      C           K      ,                                             *
*      C           PL     ,                                             *
*      C           SM     ,                                             *
*      C           H      ,                                             *
*      C           X      ,                                             *
*      C           Y1     ,                                             *
*      C           Y2     ,                                             *
*      C           TIMCON                                               *
*       COMMON                                                          *
*      C / RANGES /                                                     *
*      C           NUMANG,                                              *
*      C           ANGMAX,                                              *
*      C           DELANG,                                              *
*      C           DELRAD,                                              *
*      D           ANGINT(200)                                          *
*      D           RNGMOD(6,200)                                        *
*       COMMON                                                          *
*      C / RAYPAR /                                                     *
*      C           RANGEH,                                              *
*      D           BOTLOS(6)        ,                                   *
*      D           DRDXDC(6)        ,                                   *
*      D           PATHLN(6)        ,                                   *
*      D           RANGEC(6)        ,                                   *
*      D           SPI(6)           ,                                   *
*      D           SPT(6)           ,                                   *
*      D           TIR(6)           ,                                   *
*      D           TTR    (6)                                           *
*       COMMON                                                          *
*      C / RAYTRA /                                                     *
*      C           NCONCI,                                              *
*      C           INITLK,                                              *
*      C           Z1     ,                                             *
*      C           Z2     ,                                             *
*      C           SPVRSQ,                                              *
*      C           ANGSTR,                                              *
*      C           ANGARR,                                              *
*      C           ANGRTM,                                              *
*      C           ANGSUR,                                              *
*      C           SPDVER,                                              *
*      C           RANGET                                               *
*       COMMON                                                              *
*      C / SURDUC /                                                         *
*      C                BLA1  ,                                             *
*      C                BLA2  ,                                             *
*      C                BLA3  ,                                             *
*      C                DTRAD ,                                             *
*      C                J     ,                                             *
*      C                MSHIP ,                                             *
*      D                BF1    (128)        ,                               *
*      D                BF2    (128)        ,                               *
*      D                DELRAF(128,2),                                      *
*      D                FLN1(128)           ,                               *
*      D                FLN2(128)           ,                               *
*      D                CONLR2(40,50)       ,                               *
*      D                CONLR1(40,50)       ,                               *
*      NDUM1                                                                *
*C                                                                          *
*C      SPI( )       = SPREADING LOSS CONSTANT                      (DIMENS*
*C      SPT( )       = SPREADING LOSS CONSTANT                      (DIMENS*
*C                                                                          *
*       SPAFUN( DUMMY2 ) = ALOG10( COS( DUMMY2 )**2/A1 )                    *
*      E - 6.0                                                              *
*C                                                                          *
*       INITLK = M                                                          *
*       Z1 = DUMMY1                                                         *
*       Z2 = DUMMY2                                                         *
*       CALL                                                                *
*      S                RATRAC                                              *
*       RANGEC(N) = RANGEC(N) + RANGET                                      *
*       ANGSTR = -ANGARR                                                    *
****************************************************************************
                                     I
                                     I
                                     I
                                     I
                                     I
****************************************************************************
*       IF( Z2 .NE. DEPBOT )                                                *......O
*       * GO TO 10                                                          *      I
****************************************************************************      I
                                     I                                            I
                                     I                                            I
                                     I                                            I
****************************************************************************      I
*       BOTLOS(N) = TANH( BLA1*ANGARR ) + BLA2*ANGARR*ANGARR                *      I
****************************************************************************      I
                                     I                                            I
                                     O(...........................................O
                                     I
****************************************************************************
*    10 CONTINUE                                                            *
*       IF( Z2 .NE. DEPEVA )                                                *
*      *RETURN                                                              *
*C                                                                          *
*       DRDXDC(N) = DXDC                                                    *
*       PATHLN(N) = PL                                                      *
*       A1 = ABS( RANGEH*SIN( TIR(N) )*SIN( ANGARR )*SPDVER*DRDXDC(N) )     *
*       SPI(N) = 10.0*SPAFUN( TIR(N) )                                      *
*       SPT(N) = 10.0*SPAFUN( ANGARR )                                      *
*       TTR(N) = ANGARR                                                     *
****************************************************************************
                                     I
                                     I
                                     I
****************************************************************************
*       RETURN                                                              *
****************************************************************************


****************************************************************************
*C                                                                          *
****************************************************************************
                                     I
                                     I
                                     I
****************************************************************************
*       END                                                                 *
****************************************************************************
```

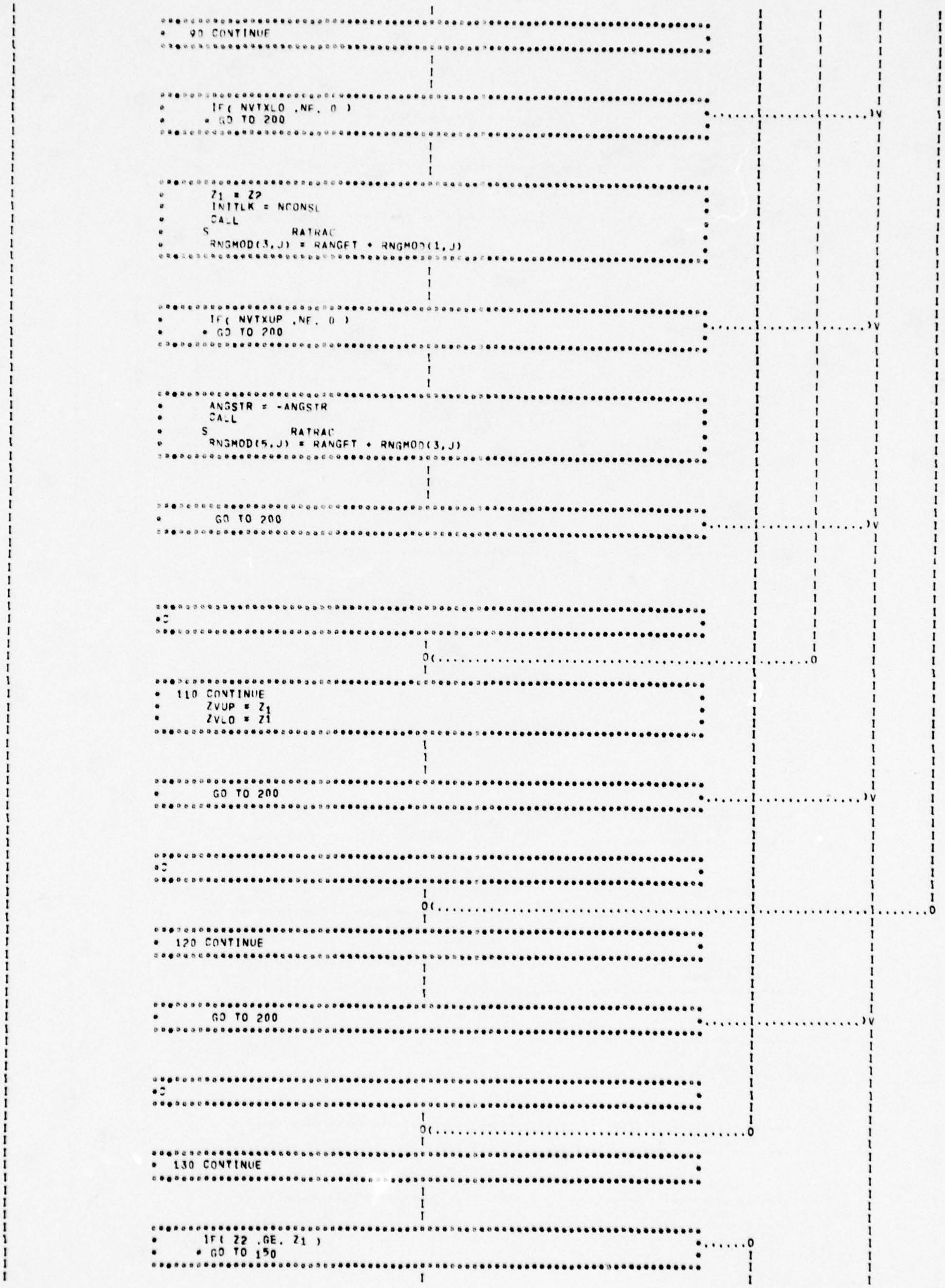
```
                                                (ENTRANCE)
                                                    I
                                                    I
                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                  •CMORPNT00        FUNCTION   MORPNT                               •
                  •CFMORP000        FUNCTION FOR POINT IN INCREASING TABLE          •
                  •      FUNCTION   MORPNI( DUMMY1, TARLES, NUMBER )                •
                  •C                                                                •
                  •C     •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                  •C     •                                                         ••
                  •C     FINDS THE PLACE IN AN INCREASING TABLE THAT BRACKETS THE DUMMY VAL•
                  •C     IF DUMMY VALUE  IS NOT WITHIN TABLE RANGE THEN             •
                  •C      MORPNT=   0 INPLIES DUMMY IS LESS THAN FIRST VALUE IN TABLE  •
                  •C      MORPNT .GT. NUMBER IMPLIES THAT DUMMY IS GREATER THAN MAXIMUM TAB•
                  •C     THE ANSWER IS THE FIRST VALUE  GREATER THAN DUMMY          •
                  •C     •                                                         ••
                  •C     •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                  •C                                                                •
                  •      DIMENSION                                                  •
                  •     D           TABLES(NUMBER)                                  •
                  •C                                                                •
                  •C     DUMMY1      = VALUE TO BE BRACKETTED                       •
                  •C     NUMBER      = NUMBER OF VALUES IN THE TABLE                •
                  •C     TABLES( )   = ARRAY OF VALUES TO BE SEARCHED               •
                  •C                                                                •
                  •      MORPNT = 0                                                 •
                  •      IF( DUMMY1 .LT. TABLES(1) )                                •
                  •     • RETURN                                                    •
                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                                                    I
                                                    I
                                                    I
                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
I.................•      DO 100                                                     •
I                 •     I           I = 2,NUMBER                                    •
I                 •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
I                                                   I
I                                                   I
I                                                   I
I                 •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
I                 •      IF( DUMMY1 .GE. TABLES(I) )                                •......O
I                 •     • GO TO 100                                                 •      I
I                 •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                                                   I                                      I
I                                                   I                                      I
I                                                   I                                      I
I                 •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                 •      MORPNT = I                                                 •      I
I                 •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                                                   I                                      I
I                                                   I                                      I
I                                                   I                                      I
I                 •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                 •      RETURN                                                     •      I
I                 •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••      I
I                                                                                          I
I                                                                                          I
I                                                   O(.....................................O
I                                                   :
I                 •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
..................•  100 CONTINUE                                                   •
                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                                                    I
                                                    I
                                                    I
                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                  •      MORPNT = NUMBER + 1                                        •
                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                                                    I
                                                    I
                                                    I
                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                  •      RETURN                                                     •
                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••

                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                  •C                                                                •
                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                                                    I
                                                    I
                                                    I
                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
                  •      END                                                        •
                  •••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••••
```

```
                                   (ENTRANCE)
                                       I
                                       I
*****************************************************************************
*CMSTRAY00         SUBROUTINE MSTRAY                                        *
*CSMSRA000         MASTER RAY-TRACING PROGRAM                               *
*CSMSTR000         SUBROUTINE FOR SETTING UP RANGE TABLE FOR RAYS           *
*       SUBROUTINE                                                          *
*      S           MSTRAY                                                   *
*C                                                                          *
*C      *********************************************************************
*C      *                                                                  **
*C      THIS IS MASTER RAY TRACING PROGRAM                                  *
*C      *                                                                  **
*C      *********************************************************************
*C                                                                          *
*       COMMON                                                              *
*      C / CONSTN /                                                         *
*      C           DEPSER,                                                  *
*      C           NCONSK,                                                  *
*      C           SPATSE,                                                  *
*      C           DEPEVA,                                                  *
*      C           NCONSL,                                                  *
*      C           SPATEV                                                   *
*       COMMON                                                              *
*      C / LCONST /                                                         *
*      C           NULAPO,                                                  *
*      C           DEPBOT,                                                  *
*      D           CONSG0(128)      ,                                       *
*      D           CONSG1(128)      .                                       *
*      D           CONSG2(128)      .                                       *
*      D           CONSV0(128)      .                                       *
*      D           DELTAZ(128)      .                                       *
*      D           DEPKYD(128)      .                                       *
*      D           SLOPEJ(128)      .                                       *
*      D           SPDKYD(128)                                              *
*       COMMON                                                              *
*      C / RANGES /                                                         *
*      C           NUANMO,                                                  *
*      C           ANGMAX,                                                  *
*      C           DELANG,                                                  *
*      C           DELRAD,                                                  *
*      D           ANGINT(200)                                              *
*      D           RNGMOD(6,200)                                            *
*       COMMON                                                              *
*      C / RAYTRA /                                                         *
*      C           NCONCJ,                                                  *
*      C           INITLK,                                                  *
*      C           Z1     ,                                                 *
*      C           Z2     ,                                                 *
*      C           SPVRSD,                                                  *
*      C           ANGSTR,                                                  *
*      C           ANGARR,                                                  *
*      C           ANGBTM,                                                  *
*      C           ANGSUR,                                                  *
*      C           SPDVER,                                                  *
*      C           RANGET                                                   *
*C                                                                          *
*C      ANGARR     = ANGLE (IN RADIANS) AT ARRIVAL POINT (Z2)               *
*C      ANGBTM     = ANGLE (IN RADIANS) OF BOTTOM BOUNCE OF RAY             *
*C      ANGMAX     = MAXIMUM ANGLE BEING CONSIDERED FOR RAYS               (*
*C      ANGSTR     = ANGLE (IN RADIANS) OF START OF RAY (Z1)                *
*C      ANGSUR     = ANGLE (IN RADIANS) OF SURFACE BOUNCE OF RAY            *
*C      CONSG0( )  = G0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C      CONSG1( )  = G1 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C      CONSG2( )  = G2 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C      CONSV0( )  = V0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C      DELSER     = DEPTH (IN K-YD) OF SEARCHER IN ITS LAYER               *
*C      DEPKYD( )  = DEPTH (IN K-YD) TO TOP OF LAYER FROM OCEAN SURFACE     *
*C      DGPRRD     = DEGREES PER RADIAN                            (DEGREES*
*C      DZVX       =                                                        *
*C      INITLK     = INITIAL VALUE OF K I.E. THE INITIAL STARTING LAYER     *
*C      NCONSK     = LAYER NUMBER OF LAYER IN WHICH SEARCHER IS FOUND       *
*C      NCONSL     = LAYER NUMBER OF LAYER IN WHICH EVADER IS FOUND         *
*C      NUANMO     = NUMBER OF ANGLES FOR RAYS MINUS ONE           (DIMENS*
*C      NUMANG     = NUMBER OF DIFFERENT ANGLES BEING CONSIDERED FOR RAYS   *
*C      NULAPO     = NUMBER OF LAYERS PLUS ONE                              *
*C      NVTXLO     = LAYER NUMBER BELOW LOWER VERTEX LAYER         (DIMENS*
*C      NVTXUP     = LAYER NUMBER IN WHICH RAY VERTEXES            (DIMENS*
*C      R          = SUMMATION OF THE DIFFERENT LAYER RANGES                *
*C      RANGET     = SUMMATION OF THE DIFFERENT LAYER RANGES                *
*C      RNGMOD( , )= RANGE (IN K-YD) FOR THE DIFFERENT MODES OF PROPAGATIO*
*C                   FIRST SUBSCRIPT IS MODE TYPE                           *
*C                   SECOND SUBSCRIPT IS RELATED TO DEPARTURE ANGLE         *
*C      SPDKYD( )  = SPEED OF SOUND PROPAGATION (IN K-YD/SEC)               *
*C      SPLASE     = PROPAGATION VELOCITY (IN K-YD/SEC) IN LAYER FOR SEARC*
*C      TI         = STARTING ANGLE (IN DEGREES) OF RAY FROM SEARCHER       *
*C      ZVLO       = DEPTH (IN K-YD) OF VERTEXING POINT                     *
*C      ZVUP       = DEPTH OF UPPER VERTEX POINT FOR RAY                    *
*C                                                                          *
*       DGPRRD = 57.2957795                                                 *
*       Z2 = DEPEVA                                                         *
*       TI = ANGMAX/DGPRRD + DELRAD                                         *
*       NUMANG = 2.0*ANGMAX/DELANG + 1.0                                    *
*****************************************************************************
                                       I
                                       I
```

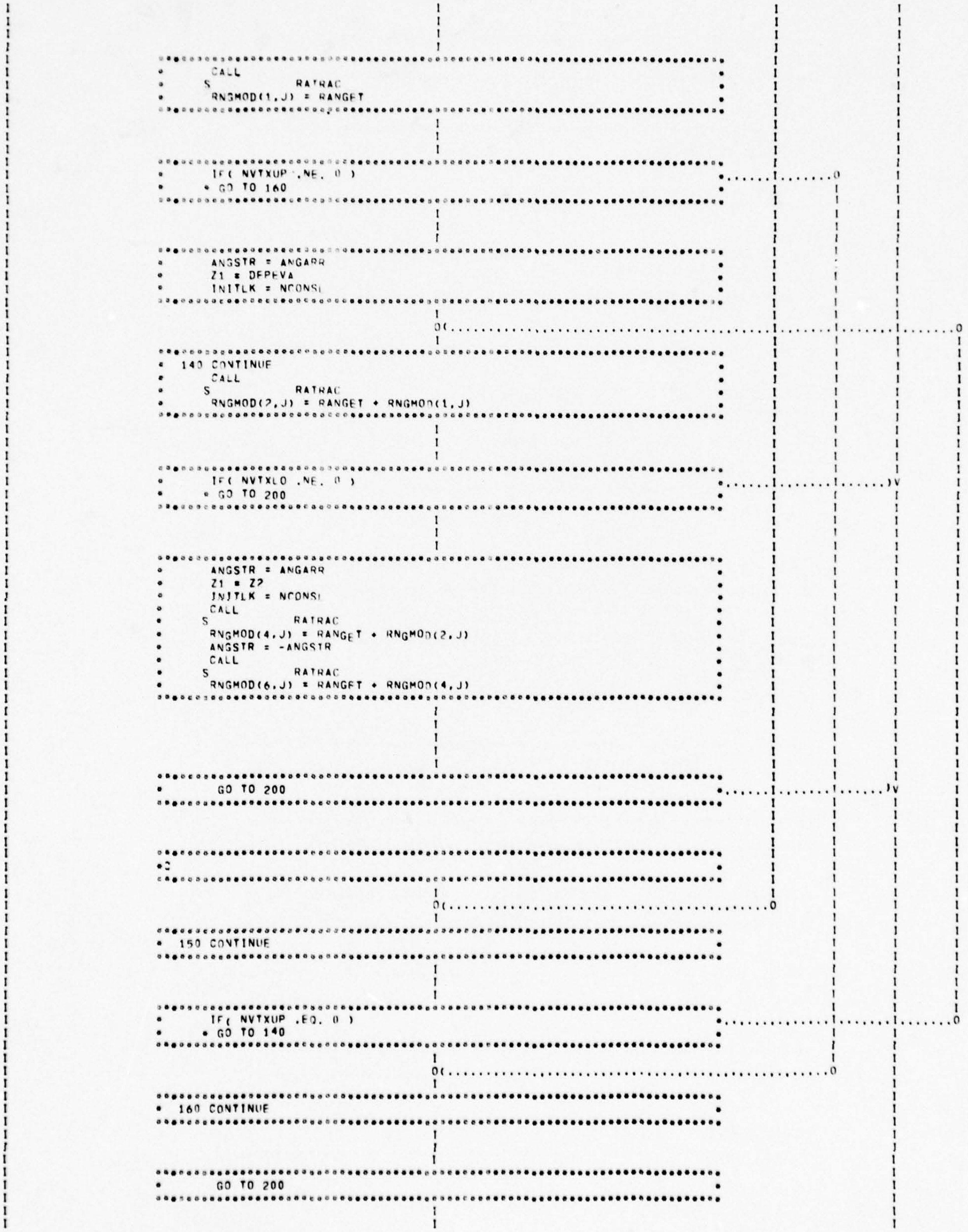


```
DO 200
I          J = 1,NUMANG

DO 100
I          I = 1,6

RNGMOD(I,J) = 0.0

100 CONTINUE

      Z1 = DEPSER
      TI = TI - DELRAD
      ANGINT(J) = TI
      ANGSTR = TI
      SPDVER = SPATSF/COS( ANGSTR )
      I = NCONSK

 10 CONTINUE

      IF( SPDVER .LE. SPDKYD(I) )
      . GO TO 20

      I = I - 1

      IF( I .NE. 0 )
      . GO TO 10

 20 CONTINUE
      NVTXUP = I
      I = NCONSK + 1

 30 CONTINUE

      IF( SPDVER .LE. SPDKYD(I) )
      . GO TO 40

      I = I + 1

      IF( I .LE. NULAPO )
      . GO TO 30
```

```
I                                     I                             I
I                                     I                             I
I        ***************************************************************  I
I        *     I = 0                                                   *  I
I        ***************************************************************  I
I                                     I                             I
I                                     O(............................O
I                                     I
I        ***************************************************************
I        *   40 CONTINUE                                               *
I        *      NVTXLO = I                                             *
I        ***************************************************************
I                                     I
I                                     I
I                                     I
I        ***************************************************************
I        *      IF( (NVTXUP .EQ. 0) .AND. (NVTXLO .EQ. 0) )            *......O
I        *      * GO TO 80                                             *      I
I        ***************************************************************      I
I                                     I                                       I
I                                     I                                       I
I                                     I                                       I
I        ***************************************************************      I
I        *      WRITE ( 6, 1000 )                                      *      I
I        ***************************************************************      I
I                                     I                                       I
I                                     I                                       I
I                                     I                                       I
I        ***************************************************************      I
I        *      IF( TI .EQ. 0.0 )                                      *......I......O
I        *      * GO TO 110                                            *      I      I
I        ***************************************************************      I      I
I                                     I                                       I      I
I                                     I                                       I      I
I                                     I                                       I      I
I                                     I                                       I      I
I        ***************************************************************      I      I
I        *      IF( NVTXUP*NVTXLO .NE. 0 )                             *......I......I......O
I        *      * GO TO 200                                            *      I      I      I
I        ***************************************************************      I      I      I
I                                     I                                       I      I      I
I                                     I                                       I      I      I
I                                     I                                       I      I      I
I        ***************************************************************      I      I      I
I        *      ZVLO = DEPROT                                          *      I      I      I
I        *      ZVUP = 0.0                                             *      I      I      I
I        *      IF( NVTXUP .NE. 0 )  CALL                              *      I      I      I
I        *     S             VERTEX                                    *      I      I      I
I        *     S ( NVTXUP, 1.0/SPDVER/SPDVER, ZVJP, Z1-DEPKYD(NVTXUP), 1 )     *      I      I      I
I        *      IF( NVTXLO .GE. 2 )  CALL                              *      I      I      I
I        *     S             VERTEX                                    *      I      I      I
I        *     S ( NVTXLO-1, 1.0/SPDVER/SPDVER, ZVLO, Z1-DEPKYD(NVTXLO-1), -1 )  *      I      I      I
I        ***************************************************************      I      I      I
I                                     I                                       I      I      I
I                                     I                                       I      I      I
I                                     I                                       I      I      I
I        ***************************************************************      I      I      I
I        *      IF( Z1 .EQ. Z2 )                                       *......I......I......I......O
I        *      * GO TO 120                                            *      I      I      I      I
I        ***************************************************************      I      I      I      I
I                                     I                                       I      I      I      I
I                                     I                                       I      I      I      I
I                                     I                                       I      I      I      I
I        ***************************************************************      I      I      I      I
I        *      IF((Z1-ZVUP)*(ZVUP-Z2).GT.0.0 .OR. (Z1-ZVLO)*(ZVLO-Z2).GT.0.0)  *......I......I......)V      I
I        *      * GO TO 200                                            *      I      I      I      I
I        ***************************************************************      I      I      I      I
I                                     I                                       I      I      I      I
I                                     O(......................................O      I      I      I
I                                     I                                              I      I      I
I        ***************************************************************             I      I      I
I        *   80 CONTINUE                                               *             I      I      I
I        *      INITLK = NCONSK                                        *             I      I      I
I        ***************************************************************             I      I      I
I                                     I                                              I      I      I
I                                     I                                              I      I      I
I                                     I                                              I      I      I
I        ***************************************************************             I      I      I
I        *      IF( TI .LT. 0.0 )                                      *......O      I      I      I
I        *      * GO TO 130                                            *      I      I      I      I
I        ***************************************************************      I      I      I      I
I                                     I                                       I      I      I      I
I                                     I                                       I      I      I      I
I                                     I                                       I      I      I      I
I        ***************************************************************      I      I      I      I
I        *      IF( Z2 .LE. Z1 )                                       *......I...O  I      I      I
I        *      * GO TO 90                                             *      I   I  I      I      I
I        ***************************************************************      I   I  I      I      I
I                                     I                                       I   I  I      I      I
I                                     I                                       I   I  I      I      I
I                                     I                                       I   I  I      I      I
I                                     I                                       I   I  I      I      I
I                                     I                                       I   I  I      I      I
I        ***************************************************************      I   I  I      I      I
I        *      CALL                                                   *      I   I  I      I      I
I        *     S             RATRAC                                    *      I   I  I      I      I
I        *      RNGMOD(I,J) = RANGET                                   *      I   I  I      I      I
I        *      ANGSTR = ANGARR                                        *      I   I  I      I      I
I        ***************************************************************      I   I  I      I      I
I                                     I                                       I   I  I      I      I
I                                     O(......................................I...O  I      I      I
```



```
    90 CONTINUE

       IF( NVTXLO .NE. 0 )
      * GO TO 200

       Z1 = Z2
       INITLK = NCONSL
       CALL
     S           RATRAC
       RNGMOD(3,J) = RANGET + RNGMOD(1,J)

       IF( NVTXUP .NE. 0 )
      * GO TO 200

       ANGSTR = -ANGSTR
       CALL
     S           RATRAC
       RNGMOD(5,J) = RANGET + RNGMOD(3,J)

        GO TO 200

C

   110 CONTINUE
       ZVUP = Z1
       ZVLO = Z1

        GO TO 200

C

   120 CONTINUE

        GO TO 200

C

   130 CONTINUE

       IF( Z2 .GE. Z1 )
      * GO TO 150
```



```
      CALL
     S              RATRAC
      RNGMOD(1,J) = RANGET

      IF( NVTXUP .NE. 0 )
     * GO TO 160

      ANGSTR = ANGARR
      Z1 = DEPEVA
      INITLK = NCONSL

  140 CONTINUE
      CALL
     S              RATRAC
      RNGMOD(2,J) = RANGET + RNGMOD(1,J)

      IF( NVTXLO .NE. 0 )
     * GO TO 200

      ANGSTR = ANGARR
      Z1 = Z2
      INITLK = NCONSL
      CALL
     S              RATRAC
      RNGMOD(4,J) = RANGET + RNGMOD(2,J)
      ANGSTR = -ANGSTR
      CALL
     S              RATRAC
      RNGMOD(6,J) = RANGET + RNGMOD(4,J)

       GO TO 200

C

  150 CONTINUE

      IF( NVTXUP .EQ. 0 )
     * GO TO 140

  160 CONTINUE

       GO TO 200
```

```
I                                               I                                                  I
I               ***************************************************************                   I
I               *C                                                            *                   I
I               ***************************************************************                   I
I                                               I                                                  I
I                                               O(.................................................O
I                                               I
I               ***************************************************************
................*  200 CONTINUE                                               *
                ***************************************************************
                                                I
                                                I
                                                I
                ***************************************************************
                *      NUANMO = NUMANG - 1                                    *
                *      NCONCI = 2                                             *
                ***************************************************************
                                                I
                                                I
                                                I
                                                I
                                                I
                ***************************************************************
                *      RETURN                                                 *
                ***************************************************************


                ***************************************************************
                *C                                                            *
                * 1000 FORMAT(                                                *
                *     F/36H THERE IS A VERTEX POINT FOR THE RAY  //)          *
                *C                                                            *
                ***************************************************************
                                                I
                                                I
                                                I
                ***************************************************************
                *      END                                                    *
                ***************************************************************
```

```
                                  (ENTRANCE)
                                      I
                                      I
****************************************************************************
*CMUEVAR00          SUBROUTINE MUEVAR                                       *
*CMUEVA000          SUBROUTINE FOR COMPUTING MU- AND VARIANCE FOR SHIPS     *
*      SUBROUTINE                                                           *
*      S            MUEVAR                                                  *
*C                                                                          *
*C     ******************************************************************** *
*C     *                                                                  **
*C     ******************************************************************** *
*C                                                                          *
*      COMMON                                                               *
*     C / CALPHA /                                                          *
*     C             ALPHAE,                                                 *
*     C             ALPHAS,                                                 *
*     C             FALPHE,                                                 *
*     C             FALPHS,                                                 *
*     C             ONMIAE,                                                 *
*     C             ONMIAS,                                                 *
*     C             STNPEV,                                                 *
*     C             STNPSE,                                                 *
*     C             TWOALE,                                                 *
*     C             TWOALS,                                                 *
*     C             NTIMEM                                                  *
*      COMMON                                                               *
*     C / SIGNAL /                                                          *
*     D             PRYSEV(128)      ,                                      *
*     D             PRYSSE(128)      ,                                      *
*     D             PRNOEV(128)      ,                                      *
*     D             PRNOSE(128)      ,                                      *
*     D             PROEVA(128)      ,                                      *
*     D             PROSER(128)      ,                                      *
*     D             VAREVA(128)      ,                                      *
*     D             VARSER(128)      ,                                      *
*     D             GMUEVA(128)      ,                                      *
*     D             GMUSER(128)      ,                                      *
*     D             DEVAEV(128)      ,                                      *
*     D             DEVASE(128)      ,                                      *
*     D             DEMUEV(128)      ,                                      *
*     D             DEMUSE(128)      ,                                      *
*     C             THREVA,                                                 *
*     C             THRSER,                                                 *
*     C             NTIMEN                                                  *
*C                                                                          *
*C     DEMUEV( )  = MODIFIED MEAN OF S/N FOR THE EVADER                     *
*C     DEMUSE( )  = MODIFIED MEAN OF S/N FOR THE SEARCHER                   *
*C     DEVAEV( )  = MODIFIED VARIANCE OF S/N FOR THE SEARCHER               *
*C     DEVASE( )  = MODIFIED VARIANCE OF S/N FOR THE SEARCHER               *
*C     FALPHE     = FRACTION  (ONMIAE**2/(1-ONMIAE**2))             (DIMENS*
*C     FALPHS     = FRACTION  (ONMIAS**2/(1-ONMIAS**2))             (DIMENS*
*C     GMUEVA( )  = MEAN OF SMOOTHED EVADER S/N                             *
*C     GMUSER( )  = MEAN OF SMOOTHED SEARCHER S/N                           *
*C     NTIMEM     = NTIMEN MINUS ONE                                (DIMENS*
*C     OMASQE     = ONE MINUS ALPHA-E   SQUARED                     (DIMENS*
*C     OMASQS     = ONE MINUS ALPHA-S   SQUARED                     (DIMENS*
*C     ONMIAE     = ONE MINUS ALPHAE                                (DIMENS*
*C     ONMIAS     = ONE MINUS ALPHAS                                (DIMENS*
*C     STNPSE     = SIGNAL TO NOISE RATIO PLUS ONE FOR THE SEARCHER         *
*C     STNPEV     = SIGNAL TO NOISE RATIO PLUS ONE FOR THE EVADER           *
*C     TWOALE     = TWO TIMES ALPHAE                                (DIMENS*
*C     TWOALS     = TWO TIMES ALPHAS                                (DIMENS*
*C     VAREVA( )  = VARIANCE OF SMOOTHED S/N FOR EVADER                     *
*C     VARSER( )  = VARIANCE OF SMOOTHED S/N FOR SEARCHER                   *
*C                                                                          *
****************************************************************************
                                      I
```

```
                                     I
                                     I
****************************************************************************
*      IF( NTIMEN .EQ. 1 )                                                 *......O
*     * GO TO 10                                                           *      I
****************************************************************************      I
                                     I                                            I
                                     I                                            I
                                     I                                            I
****************************************************************************      I
*      NTIMEM = NTIMEN - 1                                                 *      I
*      OMASQS = ONMIAS*ONMIAS                                              *      I
*      OMASQE = ONMIAE*ONMIAE                                              *      I
*      GMUSER(NTIMEN) = ALPHAS*STNPSE + ONMIAS*DEMUSE(NTIMEM)              *      I
*      GMUEVA(NTIMEN) = ALPHAE*STNPEV + ONMIAE*DEMUEV(NTIMEM)              *      I
*      VARSER(NTIMEN) = TWOALS*STNPSE*STNPSE + OMASQS*DEVASE(NTIMEM)       *      I
*      VAREVA(NTIMEN) = TWOALE*STNPEV*STNPEV + OMASQE*DEVAEV(NTIMEM)       *      I
****************************************************************************      I
                                     I                                            I
                                     I                                            I
                                     I                                            I
****************************************************************************      I
*      RETURN                                                              *      I
****************************************************************************      I
                                                                                  I
                                                                                  I
                                                                                  I
****************************************************************************      I
*C                                                                         *      I
****************************************************************************      I
                                     I                                            I
                                     O(...........................................O
                                     I
****************************************************************************
*   10 CONTINUE                                                            *
*      GMUSER(1) = ALPHAS*STNPSE + ONMIAS                                  *
*      GMUEVA(1) = ALPHAE*STNPEV + ONMIAE                                  *
*      VARSER(1) = TWOALS*( STNPSE*STNPSE + FALPHS )                       *
*      VAREVA(1) = TWOALE*( STNPEV*STNPEV + FALPHE )                       *
****************************************************************************
                                     I
                                     I
                                     I
****************************************************************************
*      RETURN                                                              *
****************************************************************************


****************************************************************************
*C                                                                         *
****************************************************************************
                                     I
                                     I
                                     I
                                     I
                                     I
****************************************************************************
*      END                                                                 *
****************************************************************************
```

```
                                   (ENTRANCE)
                                       I
                                       I
****************************************************************************
*CNTSE        NORMAL TO SEARCH EVASION                                     *
*C   ***********************************************************************
*C   *                                                                     *
*C     *NORMAL TO SEARCH EVASION                                           *
*C   *                                                                     *
*C   ***********************************************************************
*       SUBROUTINE NTSF                                                    *
*       COMMON /LAREL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSURS,STNPSF,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,FOS,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                   *
*       N1 = N - 1                                                         *
*       A1 = VXS(N1)/VYS(N1)                                               *
*       VECTXN = ( PXE(N1) - A1*PYE(N1)) /( 1.0 + A1*A1 )                  *
*       VECTYN = -A1*VECTXN                                                *
*       SQSV = SQRT (VECTXN**2+VECTYN**2)                                  *
*       DXN=VECTXN/SQSV                                                    *
*       DYN=VECTYN/SQSV                                                    *
*       VXE(N-1)=SE1*DXN                                                   *
*       VYE(N-1)=SE1*DYN                                                   *
****************************************************************************
                                       I
                                       I
                                       I
****************************************************************************
*       RETURN                                                             *
****************************************************************************



****************************************************************************
*       END                                                                *
****************************************************************************
```

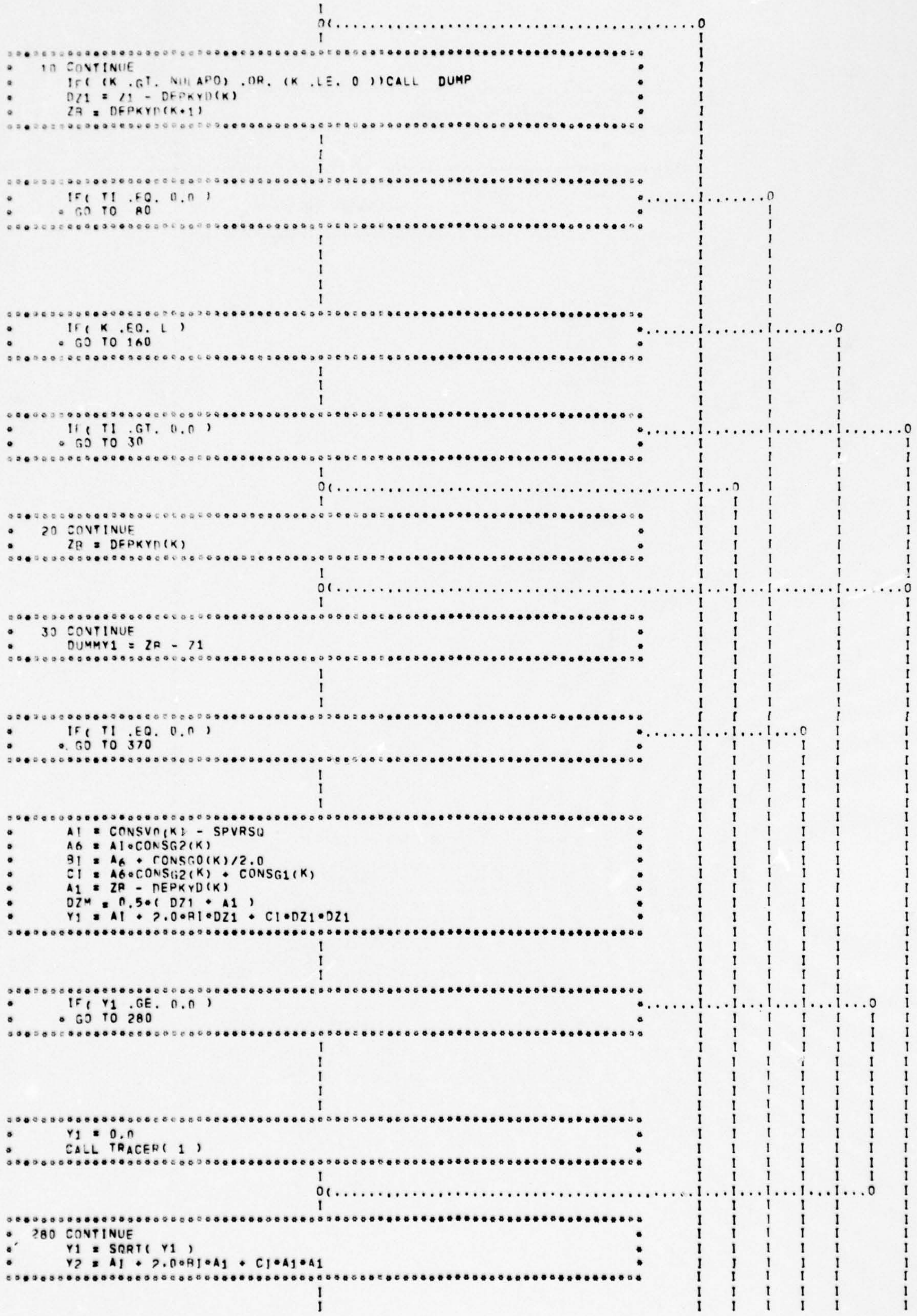
```
                           (ENTRANCE)
                               I
                               I
**************************************************************
*CPAPERF00      FUNCTION   PAPERF                            *
*CPAPER000      PAPOULIS ERROR FUNCTION ROUTINE              *
*     FUNCTION  PAPERF( DUMMY1 )                             *
*C                                                           *
*C    ********************************************************
*C    *                                                     **
*C    THIS IS THF PAPOULIS ERROR FUNCTION WHICH HAS RESULTS BETWEEN  0 A*
*C    *                                                     **
*C    ********************************************************
*C                                                           *
*C    DUMMY1    = X OF EQUATION                              *
*C              1/SQRT(2*PI) TIMES THE INTEGRAL FROM ZERO TO X  OF *
*C              EXP( -Y**2/2 )  W.R.T. Y                     *
*C              SEE PAGE 64 OF -PAPOULIS-                    *
*C                                                           *
*     PAPERF = 0.5*( 1.0 + ERF( 0.707107*DUMMY1 ) )          *
**************************************************************
                               I
                               I
                               I
**************************************************************
*     RETURN                                                 *
**************************************************************


**************************************************************
*C                                                           *
**************************************************************
                               I
                               I
                               I
**************************************************************
*     END                                                    *
**************************************************************
```

```
                                      (ENTRANCE)
                                          I
                                          I
*****************************************************************************
*CPROB          PROBABILITY DEP. ON STATE                                   *
*C    ***********************************************************************
*C    *                                                                     *
*C    *GENERATION OF PROB DEPENDING ON SHIP STATE                           *
*C    *                                                                     *
*C    ***********************************************************************
*         SUBROUTINE PROBL                                                  *
*         COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*        1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*        2NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*        3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*        4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*        5LSUBE,ALSUBS,SINPSE,SINPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*        6,F0S,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*        7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*        8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                  *
*         COMMON                                                            *
*        C / STATIC /                                                       *
*        C              NDSTAT,                                             *
*        C              NESTAT,                                             *
*        D              PEDETN(128)       ,                                 *
*        C              PERANG,                                             *
*        C              PPATHM,                                             *
*        D              PRANGE(128)       ,                                 *
*        D              PRANGS(128)       ,                                 *
*        D              PRPTEV(128)       ,                                 *
*        D              PRPTSE(128)       ,                                 *
*        D              PSDETN(128)       ,                                 *
*        C              PSRANG,                                             *
*        C              SMPEDT,                                             *
*        C              SMPSDT,                                             *
*        C              SMTCNE,                                             *
*        C              SMTCNS,                                             *
*        D              TCONEN(128)       ,                                 *
*        D              TCONSN(128)       ,                                 *
*        CDUMMYB                                                            *
*C                                                                          *
*         REAL MULT2                                                        *
*C                                                                          *
*         I=STATD                                                           *
*         J=STATE                                                           *
*         PES1=1.-PE(J)                                                     *
*         PGMULT=PGS(I)*PGE(J)                                              *
*         DIV=1.-PDS(I)*PKDS(I)*PE(J)                                       *
*         MULT2=1.-PDS(I)*PKDS(I)                                           *
*         PKILL(N)=(PGMULT*PDS(I)*PKDS(I)*PES1)/DIV                         *
*         PEVADE(N)=(PGMULT*PE(J)*MULT2)/DIV                                *
*         PPATH(N)=(PGMULT*MULT2*PES1)/DIV                                  *
*         NDSTAT = STATD                                                    *
*         NESTAT = STATE                                                    *
*         FACTOR = PGS(NDSTAT)*PGE(NESTAT)                                  *
*         PEDETN(N) = FACTOR*PDE(NESTAT)                                    *
*         PRANGE(N) = FACTOR*RANGE(N)*PDE(NESTAT)                           *
*         PRANGS(N) = FACTOR*RANGE(N)*PDS(NDSTAT)                           *
*         PRNTEV = 1.0 - PDE(NESTAT)                                        *
*         PRPTEV(N) = FACTOR*PRNTEV                                         *
*         PRNTSE = 1.0 - PDS(NDSTAT)                                        *
*         PRPTSE(N) = FACTOR*PRNTSE                                         *
*         PSDETN(N) = FACTOR*PDS(NDSTAT)                                    *
*         TCONEN(N) = PEDETN(N)*DIFTI                                       *
*         TCONSN(N) = PSDETN(N)*DIFTI                                       *
*****************************************************************************
                                          I
                                          I
                                          I
*****************************************************************************
*         RETURN                                                            *
*****************************************************************************


*****************************************************************************
*C                                                                          *
*****************************************************************************
                                          I
                                          I
                                          I
*****************************************************************************
*         END                                                               *
*****************************************************************************
```

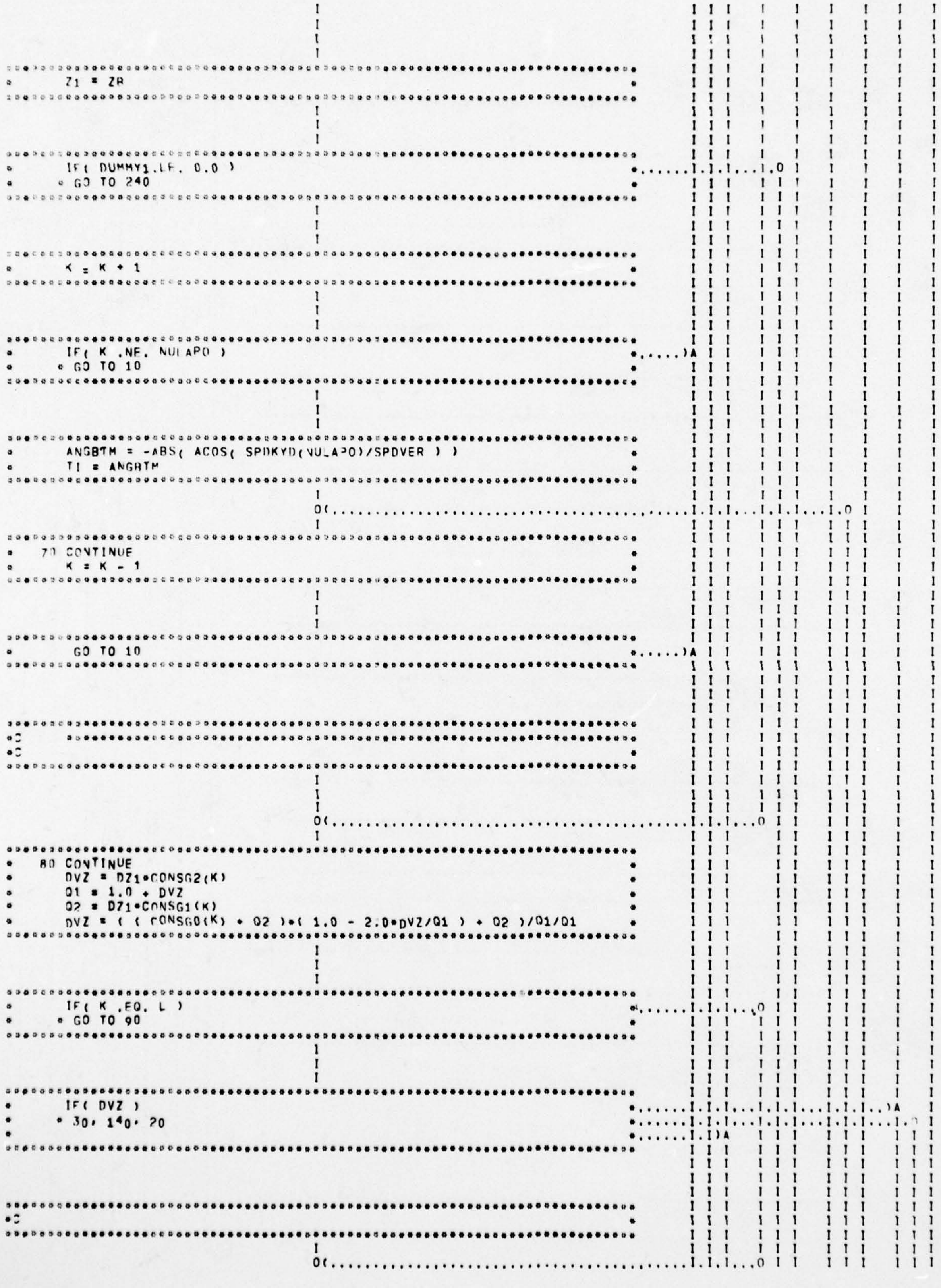
```
                              (ENTRANCE)
                                  I
                                  I
****************************************************************
*CPRBJ        UPDATING PROBABILITY                             *
*C  ************************************************************
*C  *                                                          *
*C  *  UPDATING  PROBABILITIES                                 *
*C  *                                                          *
*C  ************************************************************
*       SUBROUTINE PROBAL                                      *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,F0S,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                              *
*       COMMON                                                 *
*      C / STATIC /                                            *
*      C              NDSTAT,                                  *
*      C              NESTAT,                                  *
*      D              PEDETN(128)         ,                    *
*      C              PERANG,                                  *
*      C              PPATHM,                                  *
*      D              PRANGE(128)         ,                    *
*      D              PRANGS(128)         ,                    *
*      D              PRPTEV(128)         ,                    *
*      D              PRPTSE(128)         ,                    *
*      D              PSDETN(128)         ,                    *
*      C              PSRANG,                                  *
*      C              SMPEDT,                                  *
*      C              SMPSDT,                                  *
*      C              SMTCNE,                                  *
*      C              SMTCNS,                                  *
*      D              TCONEN(128)         ,                    *
*      D              TCONSN(128)         ,                    *
*      CDUMMYB                                                 *
*C                                                             *
*       NTIMEM = N - 1                                         *
*       PKILL(N) = PKILL(N)*PPATHM                             *
*       PPATH(N) = PPATH(N)*PPATHM                             *
*       PEVADE(N) = PEVADE(N)*PPATHM                           *
*       PEDETN(N) = PEDETN(N)*PRPTEV(NTIMEM)                   *
*       PRANGE(N) = PRANGE(N)*PRPTEV(NTIMEM)                   *
*       PRPTEV(N) = PRPTEV(N)*PRPTEV(NTIMEM)                   *
*       TCONEN(N) = TCONEN(N)*PRPTEV(NTIMEM)                   *
*       PRANGS(N) = PRANGS(N)*PRPTSE(NTIMEM)                   *
*       PRPTSE(N) = PRPTSE(N)*PRPTSE(NTIMEM)                   *
*       PSDETN(N) = PSDETN(N)*PRPTSE(NTIMEM)                   *
*       TCONSN(N) = TCONSN(N)*PRPTSE(NTIMEM)                   *
****************************************************************
                                  I
                                  I
                                  I
                                  I
                                  I
****************************************************************
*       RETURN                                                 *
****************************************************************


****************************************************************
*C                                                             *
****************************************************************
                                  I
                                  I
                                  I
****************************************************************
*       END                                                    *
****************************************************************
```

```
                                (ENTRANCE)
                                    I
                                    I
************************************************************
*CPRODET00          SUBROUTINE PRODET                      *
*CPRODE000          SUBROUTINE FOR COMPUTING THE DETECTION PROBABILITIES  *
*     SUBROUTINE                                           *
*    S              PRODET                                 *
*C                                                         *
*C    *****************************************************
*C    *                                                   **
*C    *                                                   **
*C    *****************************************************
*C                                                         *
*     COMMON                                               *
*    C / SIGNAL /                                          *
*    D              PRYSEV(128)      ,                     *
*    D              PRYSSE(128)      ,                     *
*    D              PRNOEV(128)      ,                     *
*    D              PRNOSE(128)      ,                     *
*    D              PROEVA(128)      ,                     *
*    D              PROSER(128)      ,                     *
*    D              VAREVA(128)      ,                     *
*    D              VARSER(128)      ,                     *
*    D              GMUEVA(128)      ,                     *
*    D              GMUSER(128)      ,                     *
*    D              DEVAEV(128)      ,                     *
*    D              DEVASE(128)      ,                     *
*    D              DEMUEV(128)      ,                     *
*    D              DEMUSE(128)      ,                     *
*    C              THREVA,                                *
*    C              THRSER,                                *
*    C              NTIMEN                                 *
*C                                                         *
*C    GMUEVA( )  = MEAN OF SMOOTHED EVADER S/N             *
*C    GMUSER( )  = MEAN OF SMOOTHED SEARCHER S/N           *
*C    VAREVA( )  = VARIANCE OF SMOOTHED S/N FOR EVADER     *
*C    VARSER( )  = VARIANCE OF SMOOTHED S/N FOR SEARCHER   *
*C    PROSER( )  = PROBABILITY OF DETECTION BY THE SEARCHER  *
*C    PROEVA( )  = PROBABILITY OF DETECTION BY THE EVADER  *
*C                                                         *
*     PROSER(NTIMEN) =                                     *
*    E PAPERF( ( GMUSER(NTIMEN) - THRSER )/SQRT( VARSER(NTIMEN) ) )  *
*     PROEVA(NTIMEN) =                                     *
*    E PAPERF( ( GMUEVA(NTIMEN) - THREVA )/SQRT( VAREVA(NTIMEN) ) )  *
************************************************************
                                    I
                                    I
                                    I
************************************************************
*     RETURN                                               *
************************************************************

************************************************************
*C                                                         *
************************************************************
                                    I
                                    I
                                    I
************************************************************
*     END                                                  *
************************************************************
```

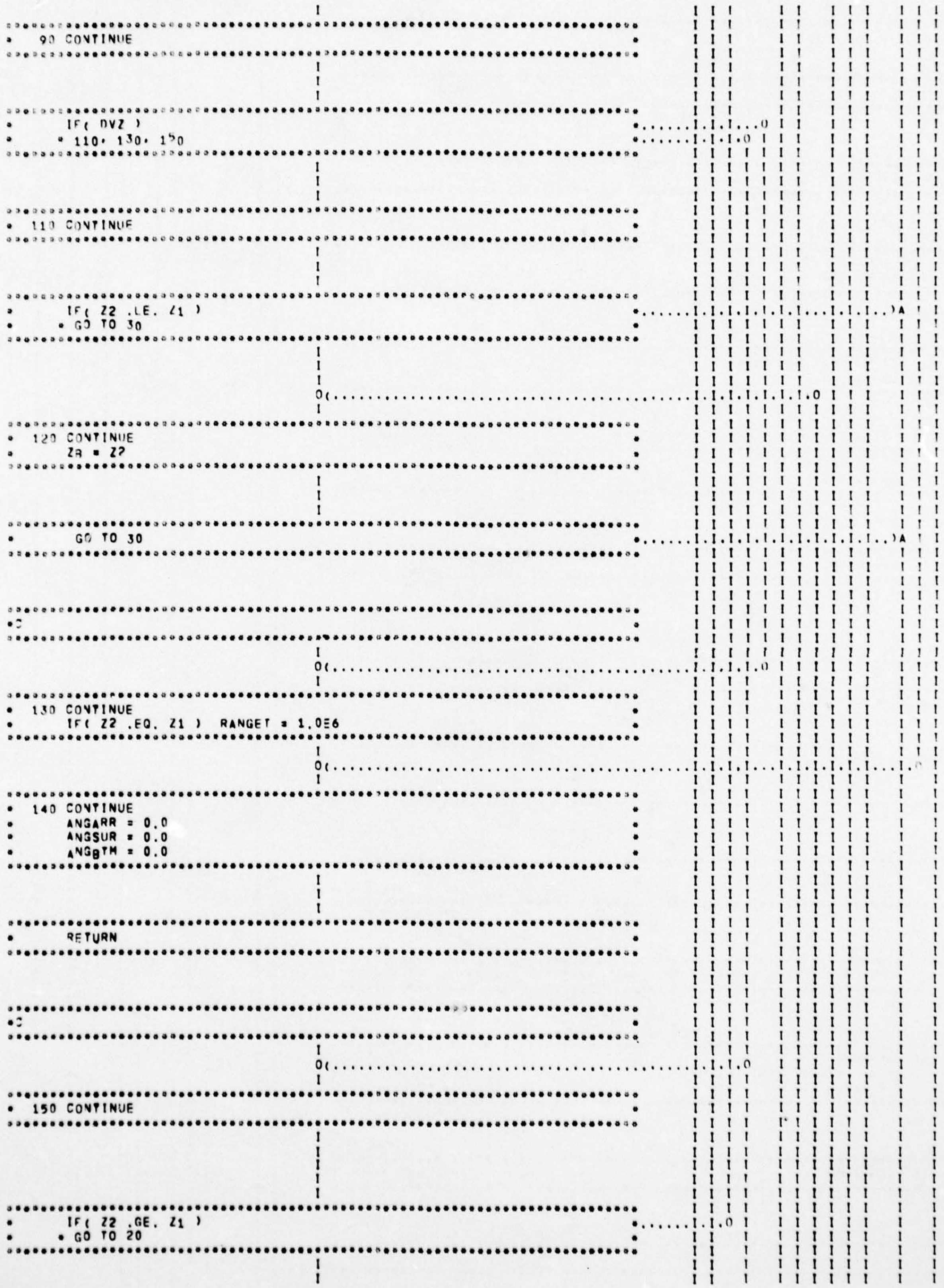
```
                                   (ENTRANCE)
                                       I
                                       I
*************************************************************************
*CRATRAC00          SUBROUTINE RATRAC                                   *
*CSRATR000          SUBROUTINE FOR RAY-TRACING                          *
*     SUBROUTINE                                                        *
*    S              RATRAC                                              *
*C                                                                      *
*C    ***************************************************************** *
*C    *                                                              ** *
*C    THIS IS THE RAY-TRACING PROGRAM                                   *
*C     ASSUMING THE RAY DOES NOT VERTEX IN THIS DEPTH RANGE             *
*C    *                                                              ** *
*C    ***************************************************************** *
*C                                                                      *
*     COMMON                                                            *
*    C / RAYTRA /                                                       *
*    C              NCONCI,                                             *
*    C              INITLK,                                             *
*    C              ZSTART,                                             *
*    C              Z2    ,                                             *
*    C              SPVRSQ,                                             *
*    C              ANGSTR,                                             *
*    C              ANGARR,                                             *
*    C              ANGBTM,                                             *
*    C              ANGSUR,                                             *
*    C              SPDVER,                                             *
*    C              RANGET                                              *
*     COMMON                                                            *
*    C / LCONST /                                                       *
*    C              NULAPO,                                             *
*    C              DEPBOT,                                             *
*    D              CONSG0(128)       .                                 *
*    D              CONSG1(128)       .                                 *
*    D              CONSG2(128)       .                                 *
*    D              CONSV0(128)       .                                 *
*    D              DELTAZ(128)       .                                 *
*    D              DEPKYD(128)       .                                 *
*    D              SLOPEJ(128)       .                                 *
*    D              SPDKYD(128)                                         *
*     COMMON                                                            *
*    C / PTHLNG /                                                       *
*    C              A1    ,                                             *
*    C              BI    ,                                             *
*    C              CDSQRD,                                             *
*    C              CI    ,                                             *
*    C              DI    ,                                             *
*    C              DXDC  ,                                             *
*    C              DZ1   ,                                             *
*    C              DZM   ,                                             *
*    C              K     ,                                             *
*    C              PL    ,                                             *
*    C              SM    ,                                             *
*    C              V     ,                                             *
*    C              X     ,                                             *
*    C              Y1    ,                                             *
*    C              Y2    ,                                             *
*    C              TIMCON                                              *
*C                                                                      *
*C     A1           = DEPTH FROM  END  OF RAY PORTION TO TOP OF LAYER   *
*C                  ALWAYS POSITIVE OR ZERO                             *
*C     ANGARR       = ANGLE (IN RADIANS) AT ARRIVAL POINT (Z2)          *
*C     ANGBTM       = ANGLE (IN RADIANS) OF BOTTOM BOUNCE OF RAY        *
*C     ANGSTR       = ANGLE (IN RADIANS) OF START OF RAY (Z1)           *
*C     ANGSUR       = ANGLE (IN RADIANS) OF SURFACE BOUNCE OF RAY       *
*C     AR     ( )   = ARRIVAL ANGLE (IN RADIANS)                        *
*C     BB     ( )   = BOTTOM BOUNCE ANGLE (IN RADIANS)                  *
*C     CONSG0( )    = G0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C     CONSG1( )    = G1 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C     CONSG2( )    = G2 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C     CONSV0( )    = V0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C     CV           = SEE -SPDVER-                                      *
*C     DEPKYD( )    = DEPTH (IN K-YD) TO TOP OF LAYER FROM OCEAN SURFACE *
*C     DZ1          = DEPTH FROM START OF RAY PORTION TO TOP OF LAYER   *
*C                  ALWAYS POSITIVE OR ZERO                             *
*C     INITLK       = INITIAL VALUE OF K I.E. THE INITIAL STARTING LAYER *
*C     J            = CONTROL SUBSCRIPT                                 *
*C     NCONCI       = CONTROL PARAMETER FOR DIFFERENT SUBPROGRAMS TO BE CAL*
*C     NULAPO       = NUMBER OF LAYERS PLUS ONE                         *
*C     RANGET( )    = SUMMATION OF THE TOTAL RANGE THROUGH THE DIFFERENT LA*
*C     SF     ( )   = SURFACE ARRIVAL ANGLE (IN RADIANS)                *
*C     SPDKYD( )    = SPEED OF SOUND PROPAGATION (IN K-YD/SEC)          *
*C     SPDVER       = VERTEX VELOCITY I.E. SPEED AT VERTEX              *
*C     T      ( )   = ANGLE (IN RADIANS)                                *
*C     Z1           = DEPTH OF START POINT OF RAY PORTION               *
*C     Z2           = DEPTH (IN K-YD) OF ENDING DEPTH                   *
*C     Z8           = DEPTH OF  END  POINT OF RAY PORTION               *
*C     ZSTART       = STARTING DEPTH (IN K-YD) OF RAY                   *
*C                                                                      *
*      ATHIRD = 1.0/3.0                                                 *
*      Z1 = ZSTART                                                      *
*      T1 = ANGSTR                                                      *
*      K = INITLK                                                       *
*      RANGET = 0.0                                                     *
*      L = MORPNT( Z2, DEPKYD, NULAPO ) - 1                             *
*      IF( (Z1 .EQ. DEPKYD(K)) .AND. (TI .LT. 0.0) )  K = K - 1         *
*      CV = SPDVER                                                      *
*      SPVRSQ = 1.0/SPDVER/SPDVER                                       *
*************************************************************************
```



```
   10 CONTINUE
      IF( (K .GT. NULAPO) .OR. (K .LE. 0 ))CALL  DUMP
      DZ1 = Z1 - DEPKYD(K)
      ZB = DEPKYD(K+1)

      IF( TI .EQ. 0.0 )
    * GO TO  80

      IF( K .EQ. L )
    * GO TO 160

      IF( TI .GT. 0.0 )
    * GO TO 30

   20 CONTINUE
      ZB = DEPKYD(K)

   30 CONTINUE
      DUMMY1 = ZB - Z1

      IF( TI .EQ. 0.0 )
    * GO TO 370

      AI = CONSV0(K) - SPVRSQ
      A6 = AI*CONSG2(K)
      BI = A6 + CONSG0(K)/2.0
      CI = A6*CONSG2(K) + CONSG1(K)
      A1 = ZB - DEPKYD(K)
      DZM = 0.5*( DZ1 + A1 )
      Y1 = AI + 2.0*BI*DZ1 + CI*DZ1*DZ1

      IF( Y1 .GE. 0.0 )
    * GO TO 280

      Y1 = 0.0
      CALL TRACER( 1 )

  280 CONTINUE
      Y1 = SQRT( Y1 )
      Y2 = AI + 2.0*BI*A1 + CI*A1*A1
      IF( Y2 .GE. 0.0 )
     * GO TO 290

      Y2 = 0.0
      CALL TRACER( 2 )

  290 CONTINUE
      Y2 = SQRT( Y2 )
      DI = DUMMY1/( Y1 + Y2 )
      A7 = DI*DI
      CDSQRD = CI*A7

      IF( ABS( CDSQRD ) .GE. 0.5 )
     * GO TO 170

      SM = ATHIRD
      POWRCD = SM

      DO 200
     I              I = 2,50

      A5 = 2*I - 1
      POWRCD = POWRCD*CDSQRD*A5/( A5 + 2.0 )
      SM = POWRCD + SM

      IF( ABS( POWRCD ) .LT. 1.0E-4 )
     * GO TO 40

  200 CONTINUE

   40 CONTINUE
      SM = SM*A7

   50 CONTINUE
      X = DI/SPDVER*( 2.0 + CONSG2(K)*( DZ1 + A1 ) +
     E ( CONSG1(K)*2.0 - CONSG0(K)*CONSG2(K) )*SM )
      RANGET = ABS( X ) + RANGET
      IF( NCONCI .NE. 1 )  CALL
     S              RAYPTH
   60 CONTINUE

      IF( Z0 .EQ. Z2 )
     * GO TO 230
```

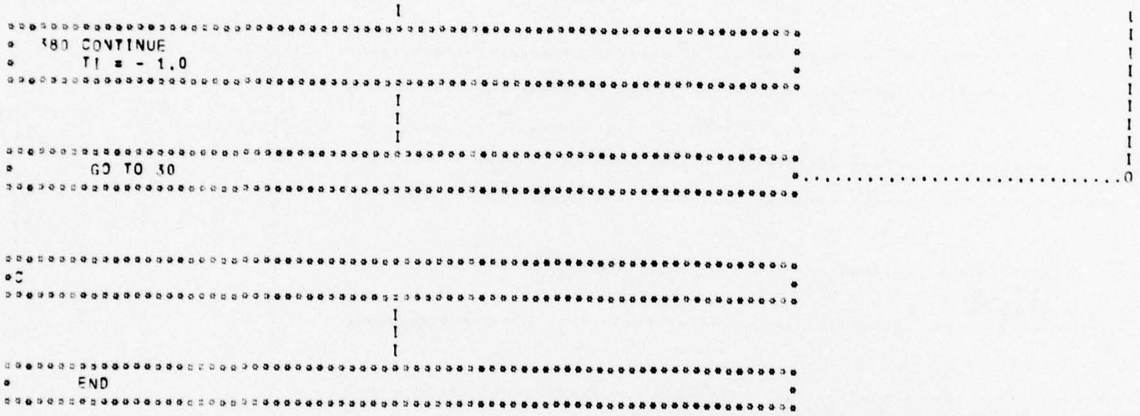


```
      Z1 = ZR

      IF( DUMMY1.LF. 0.0 )
     * GO TO 240

      K = K + 1

      IF( K .NE. NULAPO )
     * GO TO 10

      ANGBTM = -ABS( ACOS( SPDKYD(NULAPO)/SPDVER ) )
      TI = ANGBTM

   70 CONTINUE
      K = K - 1

       GO TO 10

C
C

   80 CONTINUE
      DVZ = DZ1*CONSG2(K)
      Q1 = 1.0 + DVZ
      Q2 = DZ1*CONSG1(K)
      DVZ = ( ( CONSG0(K) + Q2 )*( 1.0 - 2.0*DVZ/Q1 ) + Q2 )/Q1/Q1

      IF( K .EQ. L )
     * GO TO 90

      IF( DVZ )
     * 30, 140, 20

C
```



```
   90 CONTINUE

      IF( DVZ )
     * 110, 130, 150

  110 CONTINUE

      IF( Z2 .LE. Z1 )
     * GO TO 30

  120 CONTINUE
      ZB = Z2

      GO TO 30

C

  130 CONTINUE
      IF( Z2 .EQ. Z1 )  RANGET = 1.0E6

  140 CONTINUE
      ANGARR = 0.0
      ANGSUR = 0.0
      ANGBTM = 0.0

      RETURN

C

  150 CONTINUE

      IF( Z2 .GE. Z1 )
     * GO TO 20
```

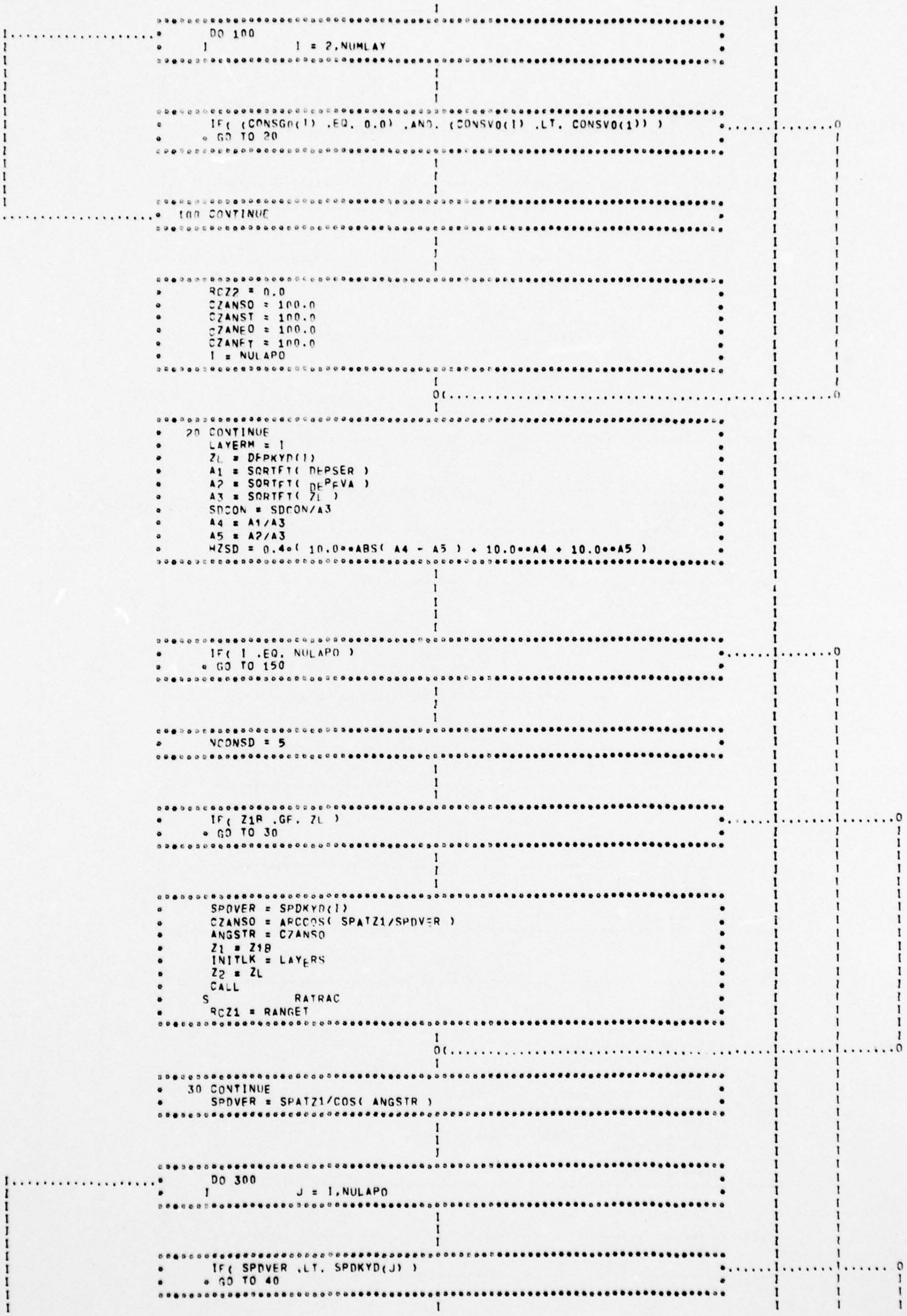
```
                                          I
*********************************************************************
*        GO TO 120                                                  *......I.I...I...I.I)A


*********************************************************************
*C                                                                  *
*********************************************************************
                                          I
                                          O(.......................................I...I...I.I.O
                                          I
*********************************************************************
*  160 CONTINUE                                                     *
*********************************************************************
                                          I
                                          I
                                          I
*********************************************************************
*       IF( RANGET .GT. 0.0 )                                       *......I.I...I...I.I)A
*      * GO TO 120                                                  *
*********************************************************************
                                          I
                                          I
                                          I
*********************************************************************
*       IF( TI .LE. 0.0 )                                           *......I.I...O
*      * GO TO 150                                                  *
*********************************************************************
                                          I
                                          I
                                          I
*********************************************************************
*       IF( Z2 .GT. Z1 )                                            *......I.I.......I.I.O
*      * GO TO 120                                                  *
*********************************************************************
                                          I
                                          I
                                          I
*********************************************************************
*        GO TO 30                                                   *......I.I.......I.I.....I.I..)A
*********************************************************************


*********************************************************************
*C                                                                  *
*********************************************************************
                                          I
                                          O(.......................................I.......I.I.....I.O
                                          I
*********************************************************************
*  170 CONTINUE                                                     *
*       A2 = SQRT( ABS( CI ) )                                      *
*       A3 = A2*DI                                                  *
*********************************************************************
                                          I
                                          I
                                          I
*********************************************************************
*       IF( CI .LE. 0.0 )                                           *......I.I.....O
*      * GO TO 190                                                  *
*********************************************************************
                                          I
                                          I
                                          I
*********************************************************************
*       H = ALOG( ABS( ( 1.0 + A3 )/( 1.0 - A3 ) ) )                *
*********************************************************************
                                          I
                                          O(.......................................I.....I.I...O
                                          I
*********************************************************************
*  180 CONTINUE                                                     *
*       SM = ( H/A2*0.5/DI - 1.0 )/CI                               *
*       V = H/A2                                                    *
*********************************************************************
                                          I
                                          I
                                          I
*********************************************************************
*        GO TO 50                                                   *......I.I.....I.I.I...I.I.....I...O
*********************************************************************


*********************************************************************
*C                                                                  *
*********************************************************************
                                          I
                                          O(.......................................I.I.....O
                              I                                   I I       I I   I I    I
*************************************************************     I I       I I   I I    I
*  190 CONTINUE                                             *     I I       I I   I I    I
*      H = 2.0*ATAN( A3 )                                   *     I I       I I   I I    I
*************************************************************     I I       I I   I I    I
                              I                                   I I       I I   I I    I
                              I                                   I I       I I   I I    I
                              I                                   I I       I I   I I    I
                              I                                   I I       I I   I I    I
                              I                                   I I       I I   I I    I
*************************************************************     I I       I I   I I    I
*       GO TO 180                                           *.....I.I.......I.I...0 I    I
*************************************************************     I I       I I     I    I
                                                                  I I       I I     I    I
*************************************************************     I I       I I     I    I
*C                                                          *     I I       I I     I    I
*************************************************************     I I       I I     I    I
                              I                                   I I       I I     I    I
                              0(..................................I.0       I I     I    I
                              I                                   I         I I     I    I
*************************************************************     I         I I     I    I
*  230 CONTINUE                                             *     I         I I     I    I
*      ANGARR = SIGN( ACOS( FVELOC( A1, K )/SPDVER ), DUMMY1 )      *     I         I I     I    I
*************************************************************     I         I I     I    I
                              I                                   I         I I     I    I
                              I                                   I         I I     I    I
*************************************************************     I         I I     I    I
*      RETURN                                               *     I         I I     I    I
*************************************************************     I         I I     I    I
                                                                  I         I I     I    I
*************************************************************     I         I I     I    I
*C                                                          *     I         I I     I    I
*************************************************************     I         I I     I    I
                              I                                   I         I I     I    I
                              0(..................................I.........0 I     I    I
                              I                                   I           I     I    I
*************************************************************     I           I     I    I
*  240 CONTINUE                                             *     I           I     I    I
*************************************************************     I           I     I    I
                              I                                   I           I     I    I
                              I                                   I           I     I    I
*************************************************************     I           I     I    I
*      IF( K .NE. 1 )                                       *.....I...........I.....)A    I
*    * GO TO 70                                             *     I           I     I    I
*************************************************************     I           I     I    I
                              I                                   I           I     I    I
                              I                                   I           I     I    I
*************************************************************     I           I     I    I
*      ANGSUR = ABS( ACOS( SPDKYD(1)/SPDVER ) )             *     I           I     I    I
*      TI = ANGSUR                                          *     I           I     I    I
*************************************************************     I           I     I    I
                              I                                   I           I     I    I
                              I                                   I           I     I    I
                              I                                   I           I     I    I
*************************************************************     I           I     I    I
*       GO TO 10                                            *.....)A          I     I    I
*************************************************************     I           I     I    I
                                                                  I           I     I    I
*************************************************************     I           I     I    I
*C                                                          *     I           I     I    I
*************************************************************     I           I     I    I
                              I                                   I           I     I    I
                              0(..................................I...........0     I    I
                              I                                   I                 I    I
*************************************************************     I                 I    I
*  370 CONTINUE                                             *     I                 I    I
*      TI = 1.0                                             *     I                 I    I
*************************************************************     I                 I    I
                              I                                   I                 I    I
                              I                                   I                 I    I
*************************************************************     I                 I    I
*      IF( DUMMY1 )                                         *.....0                 I    I
*    * 10, 70, 380                                          *.......................0    I
*************************************************************                            I
                              I                                                          I
                              I                                                          I
```



```
  380 CONTINUE
      TI = - 1.0

      GO TO 30

C

      END
```

```
                             (ENTRANCE)
                                 I
                                 I
***********************************************************************
*CRAYCTL00         SUBROUTINE RAYCTL                                  *
*CSRAYC            SUBROUTINE FOR SETTING UP PROPER RAY TABLES        *
*      SUBROUTINE                                                     *
*     S            RAYCTL                                             *
*C                                                                    *
*C     ****************************************************************
*C     *                                                             **
*C     THIS SETS UP THE REQUIRED RAY TRACE TABLES PLUS CONVERGENT ZONE VA*
*C     *                                                             **
*C     ****************************************************************
*C                                                                    *
*      COMMON                                                         *
*     C / CONSTN /                                                    *
*     C            DEPSER,                                            *
*     C            NCONSK,                                            *
*     C            SPATSE,                                            *
*     C            DEPEVA,                                            *
*     C            NCONSL,                                            *
*     C            SPATEV                                             *
*      COMMON                                                         *
*     C / LCONST /                                                    *
*     C            NULAPO,                                            *
*     C            DEPBOT,                                            *
*     D            CONSG0(128)       ,                                *
*     D            CONSG1(128)       ,                                *
*     D            CONSG2(128)       ,                                *
*     D            CONSV0(128)       ,                                *
*     D            DELTAZ(128)       ,                                *
*     D            DEPKYD(128)       ,                                *
*     D            SLOPEJ(128)       ,                                *
*     D            SPDKYD(128)                                        *
*      COMMON                                                         *
*     C / RAYTRA /                                                    *
*     C            NCONCI,                                            *
*     C            INITLK,                                            *
*     C            Z1    ,                                            *
*     C            Z2    ,                                            *
*     C            SPVRSQ,                                            *
*     C            ANGSTR,                                            *
*     C            ANGARR,                                            *
*     C            ANGBTM,                                            *
*     C            ANGSUR,                                            *
*     C            SPDVER,                                            *
*     C            RANGET                                             *
*      COMMON                                                         *
*     C / RCONST /                                                    *
*     C            ACZ   ,                                            *
*     C            AMLSRD,                                            *
*     C            BCZ   ,                                            *
*     C            HCI   ,                                            *
*     C            HZSD  ,                                            *
*     C            NCONSD,                                            *
*     C            RCZ1  ,                                            *
*     C            RCZ2  ,                                            *
*     C            SDCON ,                                            *
*     C            TCZAV1,                                            *
*     C            TCZAV2,                                            *
*     C            ZW    ,                                            *
*     NDUM6                                                           *
*      COMMON                                                         *
*     C / SURFAC /                                                    *
*     C            A6    ,                                            *
*     C            CONST2,                                            *
*     C            CONST4,                                            *
*     C            CZANGL,                                            *
*     C            CZANDL,                                            *
*     C            CZRANG,                                            *
*     C            G1SD  ,                                            *
*     C            G2SD  ,                                            *
*     C            NCZRAS,                                            *
*     C            NZONE ,                                            *
*     C            RSD   ,                                            *
*     C            RSD1  ,                                            *
*     C            SCSD  ,                                            *
*     C            SQRTZL,                                            *
*     C            SS    ,                                            *
*     C            ZL    ,                                            *
*     CDUMMYZ                                                         *
*C                                                                    *
*C     AMLSRD      = CONSTANT REQQUIRED FOR AMOS CALCULATIONS         *
*C     ANGARR      = ANGLE (IN RADIANS) OF ARRIVAL AT Z2              *
*C     ANGBTM      = ANGLE (IN RADIANS) OF BOTTOM BOUNCE OF RAY       *
*C     ANGSTR      = ANGLE (IN RADIANS) OF START OF RAY (Z1)          *
*C     ANGSUR      = ANGLE (IN RADIANS) OF SURFACE BOUNCE OF RAY      *
*C     CONSG0( )   = G0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C     CONSG1( )   = G1 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C     CONSG2( )   = G2 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C     CONSV0( )   = V0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C     COSZ1B = COSINE OF STARTING ANGLE OF RAY                  (DIMENS*
*C     CUB         =                                                  *
*C     CUBCAL      =                                                  *
*C     CZANEO      = SMALLER CONVERGENT ZONE ANGLE (IN RADIANS) FOR EVADER*
*C     CZANET      = LARGER  CONVERGENT ZONE ANGLE (IN RADIANS) FOR EVADER*
*C     CZANSO      = SMALLER CONVERGENT ZONE ANGLE (IN RADIANS) FOR SEARCH*
*C     CZANST      = LARGER  CONVERGENT ZONE ANGLE (IN RADIANS) FOR SEARCH*
*C     DEPKYD( )   = DEPTH (IN K-YD) TO TOP OF LAYER FROM OCEAN SURFACE *
*C     DPTOZ1      = DEPTH (IN K-YD) FROM TOP OF LAYER TO Z1          *
*C     DZVX        = DEPTH BELOW LAYER (IN K-YD) OF POINT OF VERTEX   *
```

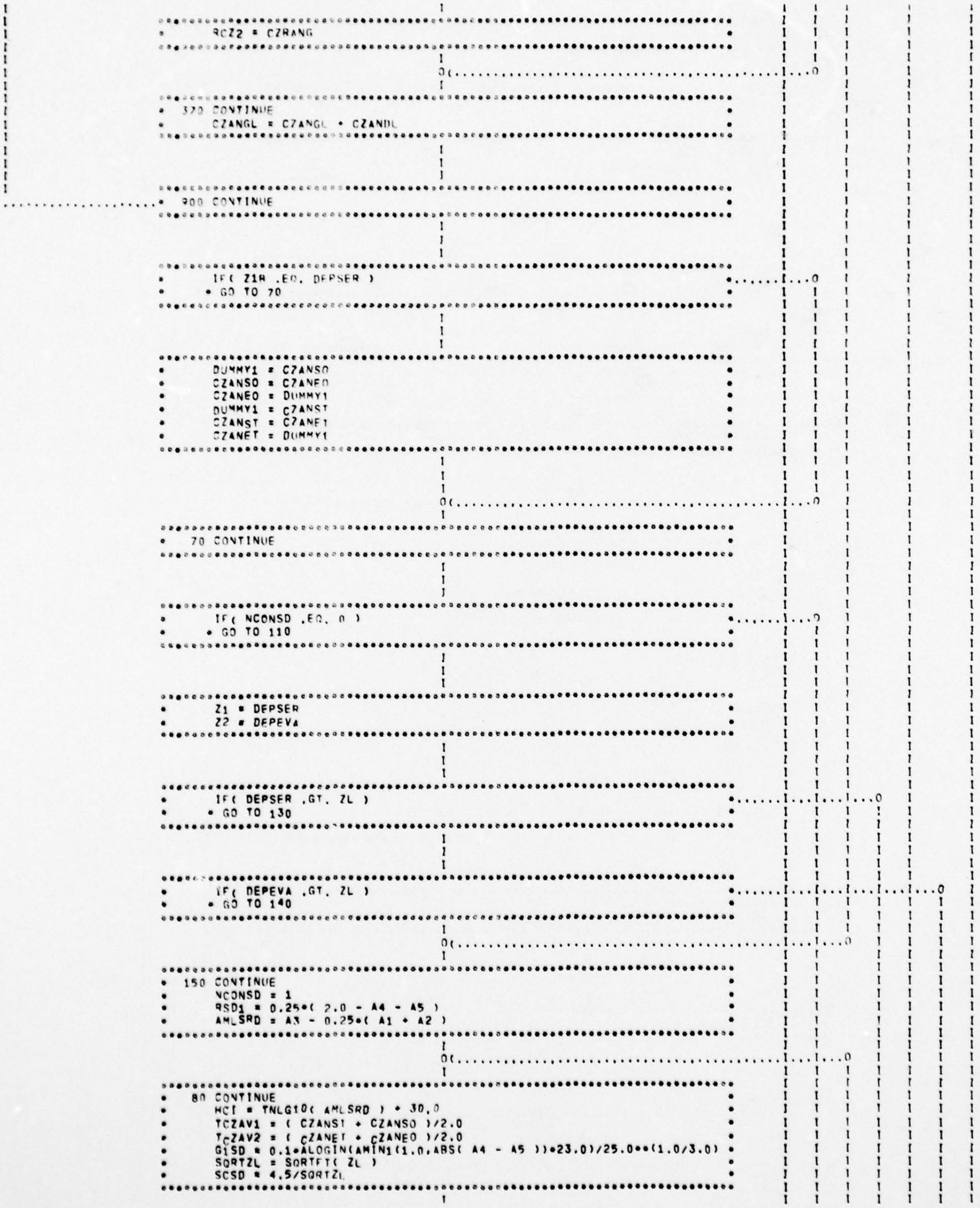
```
*C     IVMAX       =                                                     *
*C     KB          = LAYER WHICH CONTAINS Z1B                            *
*C     LAYERL      = LAYER IN WHICH A LOWER VERTEX IS FOUND      (DIMENS*
*C     LAYERM      = LAYER IN WHICH VELOCITY PROFILE HAS A ZERO  (DIMENS*
*C     LAYERS      = LAYER NUMBER OF HIGHER DEPTH OF SHIP        (DIMENS*
*C     NCONCI      = CONTROL PARAMETER FOR DIFFERENT SUBROUTINES TO BE CAL*
*C     NCONSD      = AMOS CONSTANT SD                                    *
*C     NCONZ1      = LAYER NUMBER WHICH CONTAINS Z1                      *
*C     NULAPO      = NUMBER OF LAYERS PLUS ONE                           *
*C     RANGET      = TOTAL RANGE (IN K-YD) AS COMPUTED BY RAY-TRACING SUBR*
*C     RCZ1        =                                                     *
*C     SDCON       = WAVE HEIGTH PARAMETER                               *
*C     SPATEV      = PROPAGATION SPEED AT EVADER                       (K*
*C     SPATSE      = PROPAGATION SPEED AT SEARCHER                       *
*C     SPATZ1      = PROPAGATION SPEED (IN K-YD/SEC) AT POINT STARTING POI*
*C     SPDKYD( )   = SPEED OF SOUND PROPAGATION (IN K-YD/SEC)            *
*C     SQRTFT      = FUNCTION TO COMPUTE THE SQUARE ROOT OF K-YD IN FEET *
*C     TCZAV1      =           AVERAGE OF CONVERGENT ZONE ANGLES        (*
*C     TCZAV2      =           AVERAGE OF CONVERGENT ZONE ANGLES        (*
*C     TNLG10      = FUNCTION TO CONVERT DIMENSIONLESS QUANTITY TO DB    *
*C     XSD         = SEE -AMLSRD-                                     (SQR*
*C     Z1          = STARTING DEPTH (IN K-YD) OF RAY                     *
*C     Z2          = ENDING DEPTH (IN K-YD) OF RAY                       *
*C     Z2B         = ENDING DEPTH (IN K-YD) OF CONVERGENT ZONE RAY       *
*C     ZL          = DEPTH OF MAX OR MIN POINT IN VELOCITY PROFILE       *
*C     ZVLO        = DEPTH (IN K-YD) OF VERTEX POINT OF CONVERGENT RAY   *
*C                                                                       *
*      NAMELIST / SEARCH / CZANGL, CZRANG, Z1                            *
*C                                                                       *
*      SQRTFT( DUMMY1 ) = SQRT( 3000.0*DUMMY1 )                          *
*C                                                                       *
*      RCZ1 = 0.0                                                        *
*      NCONCI = 1                                                        *
*      SDCON = 4.5                                                       *
*      IF( ZW .GT. 4.0 )  SDCON = 2.0*SDCON                              *
*      NCONSK = MORPNT( DEPSER, DEPKYD, NULAPO ) - 1                     *
*      SPATSE = FVELOC( DEPSER - DEPKYD(NCONSK), NCONSK )                *
*      NCONSL = MORPNT( DEPEVA, DEPKYD, NULAPO ) - 1                     *
*      SPATEV = FVELOC( DEPEVA - DEPKYD(NCONSL), NCONSL )                *
*      Z2 = AMAX1( DEPSER, DEPEVA )                                      *
*      Z1 = AMIN1( DEPSER, DEPEVA )                                      *
*      SPATZ1 = SPATSE                                                   *
*      LAYERS = NCONSK                                                   *
**************************************************************************
                                    I
                                    I
                                    I
**************************************************************************
*      IF( DEPEVA .GE. DEPSER )                                          *......O
*     * GO TO  10                                                        *      I
**************************************************************************      I
                                    I                                           I
                                    I                                           I
                                    I                                           I
**************************************************************************      I
*      SPATZ1 = SPATEV                                                   *      I
*      LAYERS = NCONSL                                                   *      I
**************************************************************************      I
                                    I                                           I
                                    O(..........................................O
                                    I
**************************************************************************
*   10 CONTINUE                                                          *
*      COSZ1B = SPATZ1/SPDKYD(NULAPO)                                    *
**************************************************************************
                                    I
                                    I
                                    I
**************************************************************************
*      IF( COSZ1B .GE. 1.0 )                                             *......O
*     * GO TO 90                                                         *      I
**************************************************************************      I
                                    I                                           I
                                    I                                           I
                                    I                                           I
**************************************************************************      I
*      Z1B = Z1                                                          *      I
*      Z2B = Z2                                                          *      I
*      SPDVER = SPDKYD(NULAPO)                                           *      I
*      CZANST = ACOS( COSZ1B )                                           *      I
*      ANGSTR = CZANST                                                   *      I
*      INITLK = LAYERS                                                   *      I
*      Z2 = DEPKYD(NULAPO)                                               *      I
*      CALL                                                              *      I
*     S            RATRAC                                                *      I
*      RCZ2 = RANGET                                                     *      I
*      ANGSTR = -ANGARR                                                  *      I
*      Z1 = Z2                                                           *      I
*      INITLK = NULAPO                                                   *      I
*      Z2 = Z2B                                                          *      I
*      CALL                                                              *      I
*     S            RATRAC                                                *      I
*      RCZ2 = RANGET + RCZ2                                              *      I
*      CZANET = ABS( ANGARR )                                            *      I
*      NUMLAY = NULAPO - 1                                               *      I
**************************************************************************      I
                                    I                                           I
                                    I                                           I
```



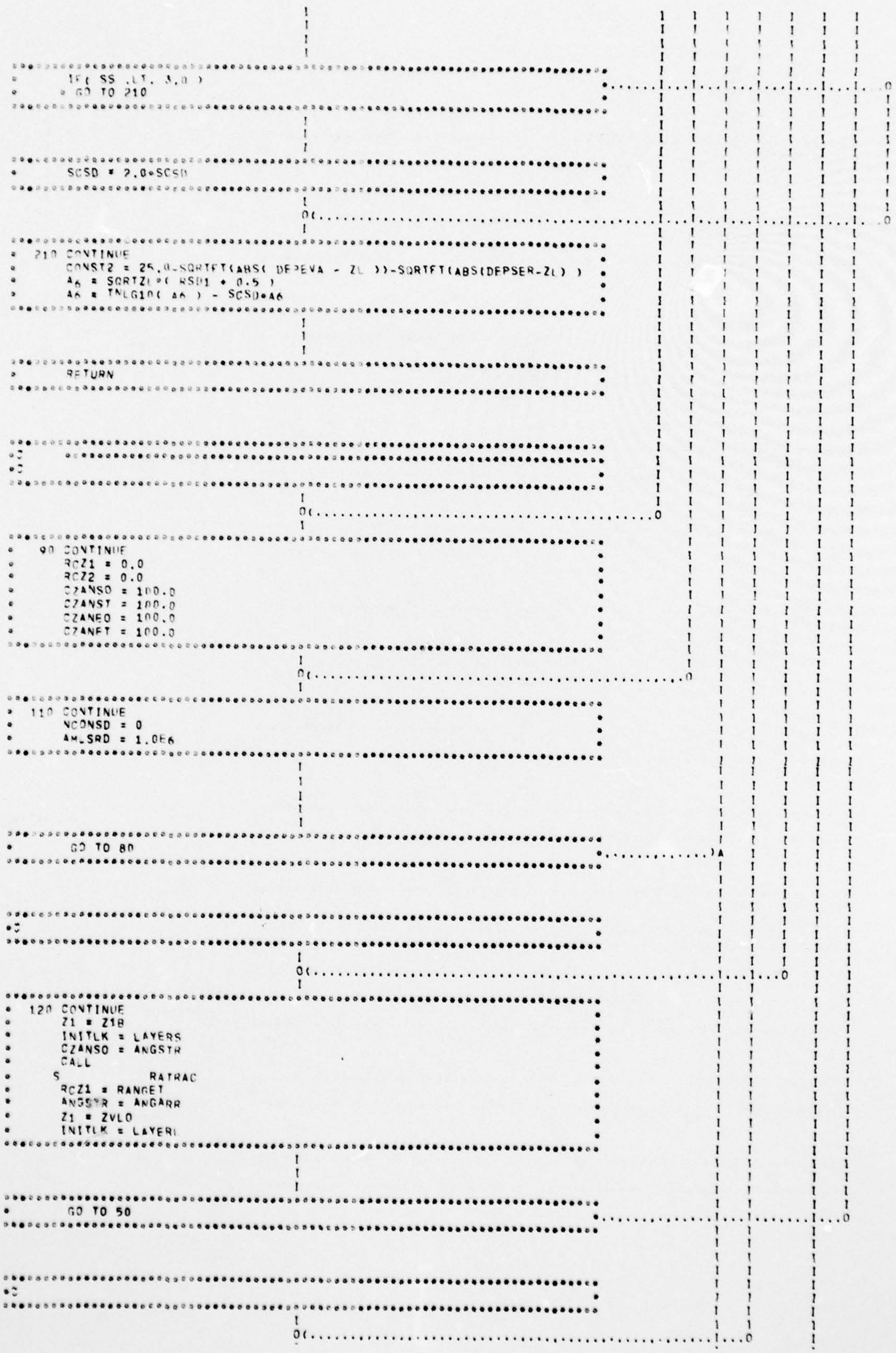
```
      DO 100
     I              I = 2,NUMLAY

      IF( (CONSGO(I) .EQ. 0.0) .AND. (CONSVO(I) .LT. CONSVO(1)) )
     * GO TO 20

  100 CONTINUE

      RCZ2 = 0.0
      CZANSO = 100.0
      CZANST = 100.0
      CZANEO = 100.0
      CZANET = 100.0
      I = NULAPO

   20 CONTINUE
      LAYERM = I
      ZL = DEPKYD(I)
      A1 = SQRTFT( DEPSER )
      A2 = SQRTFT( DEPEVA )
      A3 = SQRTFT( ZL )
      SDCON = SDCON/A3
      A4 = A1/A3
      A5 = A2/A3
      HZSD = 0.4*( 10.0**ABS( A4 - A5 ) + 10.0**A4 + 10.0**A5 )

      IF( I .EQ. NULAPO )
     * GO TO 150

      NCONSD = 5

      IF( Z1B .GE. ZL )
     * GO TO 30

      SPDVER = SPDKYD(I)
      CZANSO = ARCCOS( SPATZ1/SPDVER )
      ANGSTR = CZANSO
      Z1 = Z1B
      INITLK = LAYERS
      Z2 = ZL
      CALL
     S              RATRAC
      RCZ1 = RANGET

   30 CONTINUE
      SPDVER = SPATZ1/COS( ANGSTR )

      DO 300
     I              J = I,NULAPO

      IF( SPDVER .LT. SPDKYD(J) )
     * GO TO 40
```

```
I                                         I
I                                         I
I                                         I
I                                         I
I                                         I
                  ****************************************************************
.................*  300 CONTINUE                                                  *
                  ****************************************************************
                                          I
                                          I
                                          I
                  ****************************************************************
                  *       GO TO 90                                               *.....)V
                  ****************************************************************

                  ****************************************************************
                  *C                                                             *
                  ****************************************************************
                                          I
                                          0(...........................................0
                                          I
                  ****************************************************************
                  *   40 CONTINUE                                                *
                  *      LAYERL = J - 1                                          *
                  *      CALL                                                    *
                  *     S               VERTEX                                   *
                  *     S ( LAYFRL, 1.0/SPDVER/SPDVER, ZVLO, Z1 - DEPKYD(LAYERL), -1 )  *
                  *      Z2 = ZVLO                                               *
                  *      ANGSTR = 0.005                                          *
                  ****************************************************************
                                          I
                                          I
                                          I
                  ****************************************************************
                  *       IF( NCONSD .EQ. 0 )                                    *.....................0
                  *     * GO TO 120                                              *
                  ****************************************************************
                                          I
                                          I
                                          I
                  ****************************************************************
                  *       Z1 = ZL                                                *
                  *       INITLK = LAYERM                                        *
                  *       CALL                                                   *
                  *     S               RATRAC                                   *
                  *      RCZ1 = RANGET + RCZ1                                    *
                  *      ANGSTR = 0.0                                            *
                  *      Z1 = ZVLO                                               *
                  *      INITLK = LAYERL                                         *
                  ****************************************************************
                                          I
                                          I
                                          I
                  ****************************************************************
                  *       IF( Z2B .GE. ZL )                                      *.............................0
                  *     * GO TO 50                                               *
                  ****************************************************************
                                          I
                                          I
                                          I
                                          I
                                          I
                  ****************************************************************
                  *       RCZ1 = RANGET + RCZ1                                   *
                  *       ANGSTR = -0.005                                        *
                  *       Z1 = ZL                                                *
                  *       INITLK = LAYERM                                        *
                  ****************************************************************
                                          I
                                          0(...................................................0
                                          I
                  ****************************************************************
                  *   50 CONTINUE                                                *
                  *      Z2 = Z2B                                                *
                  *      CALL                                                    *
                  *     S               RATRAC                                   *
                  *      RCZ1 = RANGET + RCZ1                                    *
                  *      CZANEO = ABS( ANGARR )                                  *
                  *   60 CONTINUE                                                *
                  *C                                                             *
                  *C     DO SEARCH TO FIND MAXIMUM AND MINIMUM RANGES            *
                  *C                                                             *
                  *      NCZRAS = 100                                            *
                  *      CZANDL = ( CZANST - CZANSO )/FLOAT( NCZRAS )            *
                  *      CZANGL = CZANSO                                         *
                  *C                                                             *
                  *      NCZRAS = NCZRAS - 1                                     *
                  ****************************************************************
                                          I
                                          I
                                          I
                  ****************************************************************
I................*      DO 900                                                  *
I                 *    I          N = 1,NCZRAS                                   *
I                 ****************************************************************
I                                         I
```

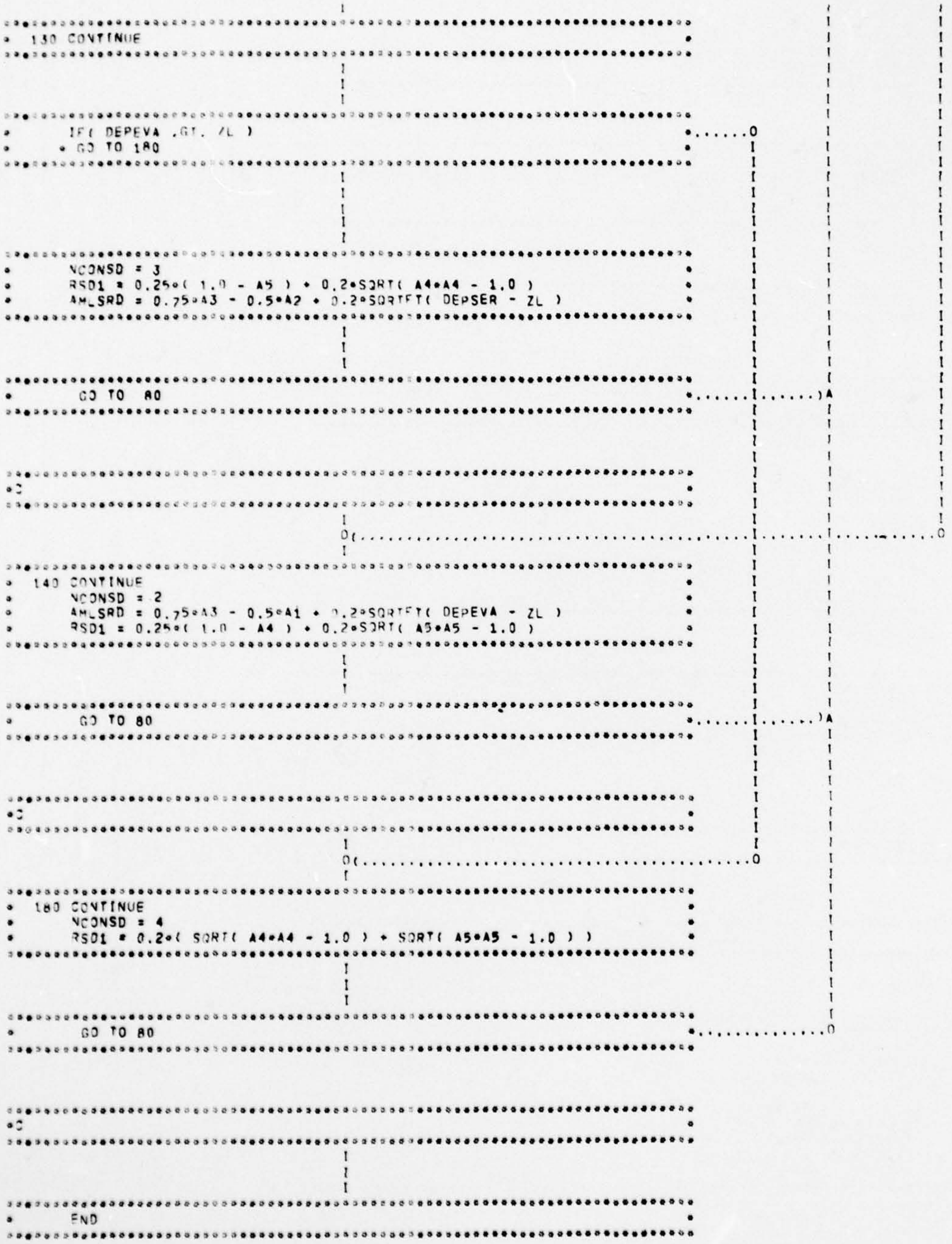
```
      SPDVER = SPATZ1/COS( CZANGL )
      SPVRSQ = 1.0/SPDVER/SPDVER
C
C     FIND LAYER IN WHICH RAY VERTEXES

      DO 800
  I            J = LAYERS,NULAPO

      IF( SPDVER .LT. SPDKYD(J) )
    * GO TO 390

  800 CONTINUE

      CALL TRACER( 39)

  390 CONTINUE
C
C     FIND DEPTH AT WHICH RAY VERTEXES
C
      CALL
     S              VERTEX
     S ( J-1, SPVRSQ,      Z2, Z1B - DEPKYD(J-1), -1 )
C
C     COMPUTE HORIZONTAL TRAVEL OF RAY
C
      Z2 = Z2 - 1.0E-6
      ANGSTR = CZANGL
      Z1 = Z1B
      CALL
     S              RATRAC
      CZRANG = RANGET
      ANGSTR = -ANGARR
      Z1 = Z2
      Z2 = Z2B
      CALL
     S              RATRAC
      CZRANG = CZRANG + RANGET
C
C     CHECK TO SEE IF THIS NEW RANGE IS A MAX OR MIN
C

      IF( CZRANG .GE. RCZ1 )
    * GO TO 380

      RCZ1 = CZRANG

  380 CONTINUE

      IF( CZRANG .LE. RCZ2 )
    * GO TO 370
```



```
      RCZ2 = CZRANG

  370 CONTINUE
      CZANGL = CZANGL + CZANDL

  900 CONTINUE

      IF( Z1B .EQ. DEPSER )
     * GO TO 70

      DUMMY1 = CZANSO
      CZANSO = CZANEO
      CZANEO = DUMMY1
      DUMMY1 = CZANST
      CZANST = CZANET
      CZANET = DUMMY1

   70 CONTINUE

      IF( NCONSD .EQ. 0 )
     * GO TO 110

      Z1 = DEPSER
      Z2 = DEPEVA

      IF( DEPSER .GT. ZL )
     * GO TO 130

      IF( DEPEVA .GT. ZL )
     * GO TO 140

  150 CONTINUE
      NCONSD = 1
      RSD1 = 0.25*( 2.0 - A4 - A5 )
      AMLSRD = A3 - 0.25*( A1 + A2 )

   80 CONTINUE
      HCI = TNLG10( AMLSRD ) + 30.0
      TCZAV1 = ( CZANST + CZANSO )/2.0
      TCZAV2 = ( CZANET + CZANEO )/2.0
      G1SD = 0.1*ALOGIN(AMIN1(1.0,ABS( A4 - A5 ))*23.0)/25.0**(1.0/3.0)
      SQRTZL = SQRTFT( ZL )
      SCSD = 4.5/SQRTZL
```



```
      IF( SS .LT. 3.0 )
     * GO TO 210

      SCSD = 2.0*SCSD

  210 CONTINUE
      CONST2 = 25.0-SQRTFT(ABS( DEPEVA - ZL ))-SQRTFT(ABS(DEPSER-ZL) )
      A6 = SQRTZL*( RSD1 + 0.5 )
      A6 = TNLG10( A6 ) - SCSD*A6

      RETURN

C
C

   90 CONTINUE
      RCZ1 = 0.0
      RCZ2 = 0.0
      CZANSO = 100.0
      CZANST = 100.0
      CZANEO = 100.0
      CZANET = 100.0

  110 CONTINUE
      NCONSD = 0
      AMLSRD = 1.0E6

      GO TO 80

C

  120 CONTINUE
      Z1 = Z1B
      INITLK = LAYERS
      CZANSO = ANGSTR
      CALL
     S            RATRAC
      RCZ1 = RANGET
      ANGSTR = ANGARR
      Z1 = ZVLO
      INITLK = LAYERL

      GO TO 50

C
```



```
  130 CONTINUE

      IF( DEPEVA .GT. ZL )
     * GO TO 180

      NCONSD = 3
      RSD1 = 0.25*( 1.0 - A5 ) + 0.2*SQRT( A4*A4 - 1.0 )
      AMLSRD = 0.75*A3 - 0.5*A2 + 0.2*SQRTFT( DEPSER - ZL )

      GO TO  80

C

  140 CONTINUE
      NCONSD = 2
      AMLSRD = 0.75*A3 - 0.5*A1 + 0.2*SQRTFT( DEPEVA - ZL )
      RSD1 = 0.25*( 1.0 - A4 ) + 0.2*SQRT( A5*A5 - 1.0 )

      GO TO 80

C

  180 CONTINUE
      NCONSD = 4
      RSD1 = 0.2*( SQRT( A4*A4 - 1.0 ) + SQRT( A5*A5 - 1.0 ) )

      GO TO 80

C

      END
```

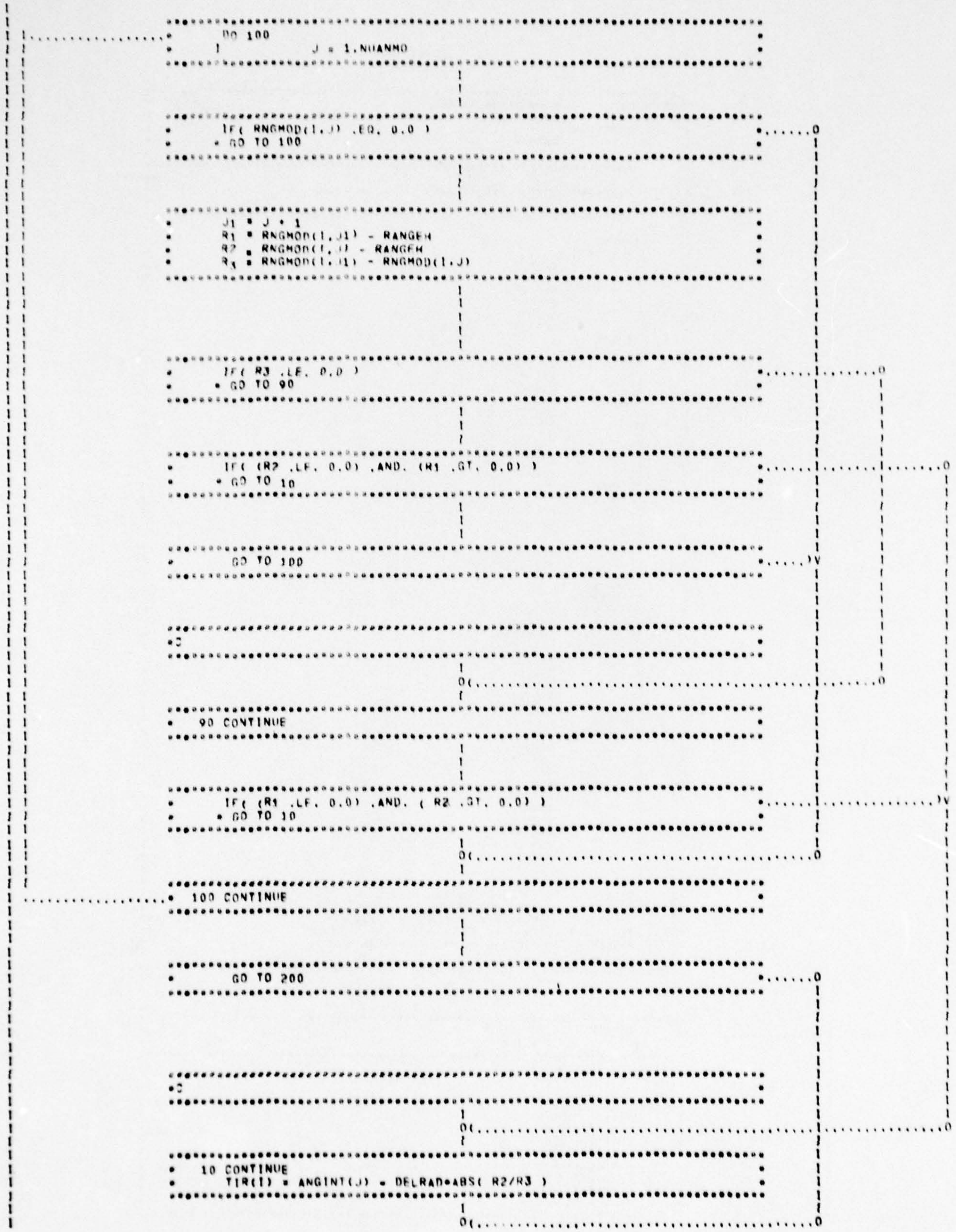
AD-A068 881 GENERAL ELECTRIC CO SYRACUSE N Y HEAVY MILITARY ELEC--ETC F/G 17/1
SONAR SIMULATION COMPUTER PROGRAMS. VOLUME 2. FLOW CHARTS.(U)
NOV 68 N00140-68-C-0372

UNCLASSIFIED NL

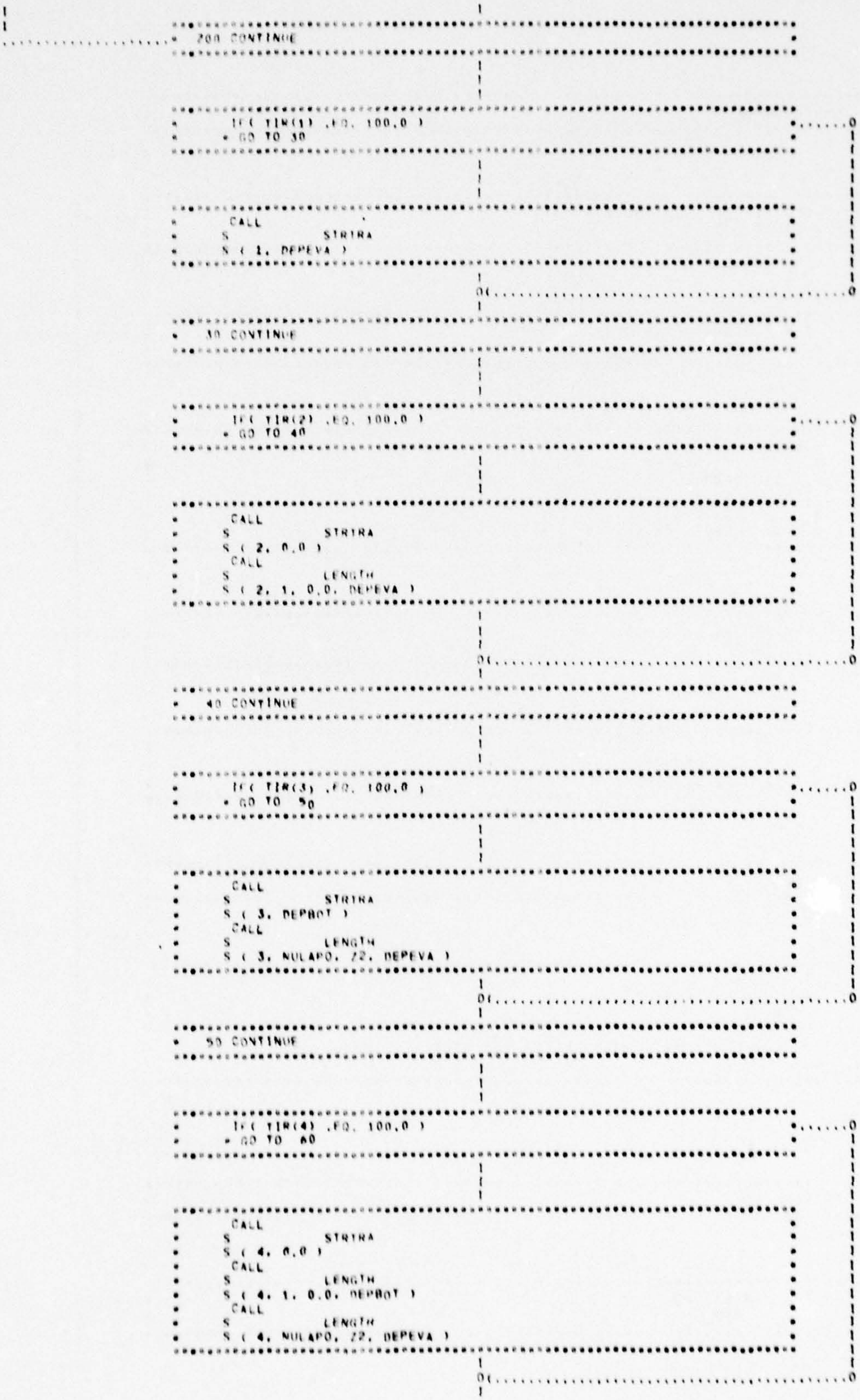
2 OF 2
AD
A068881

END
DATE
FILMED
7-79
DDC

```
                                                  (ENTRANCE)
                                                       I
                                                       I
                  ****************************************************************************
                  *CRAYNOW00          SUBROUTINE RAYNOW                                      *
                  *CSRAVD0D0          SUBROUTINE FOR FINDING POSSIBLE RAYS AND PARAMETERS    *
                  *     SUBROUTINE                                                           *
                  *    S              RAYNOW                                                 *
                  *C                                                                         *
                  *C     ********************************************************************
                  *C     *                                                                  **
                  *C     THIS FINDS RAYS CLOSE TO RANGE BETWEEN SHIPS                        *
                  *C     *                                                                  **
                  *C     ********************************************************************
                  *C                                                                         *
                  *      COMMON                                                              *
                  *     C / CONSTN /                                                         *
                  *     C             DEPSER,                                                *
                  *     C             NCONSK,                                                *
                  *     C             SPATSF,                                                *
                  *     C             DEPEVA,                                                *
                  *     C             NCONSL,                                                *
                  *     C             SPATEV                                                 *
                  *      COMMON                                                              *
                  *     C / LCONST /                                                         *
                  *     C             NULAPO,                                                *
                  *     C             DEPHOT,                                                *
                  *     D             CONSG0(128)       ,                                    *
                  *     D             CONSG1(128)       ,                                    *
                  *     D             CONSG2(128)       ,                                    *
                  *     D             CONSV0(128)       ,                                    *
                  *     D             DELTAZ(128)       ,                                    *
                  *     D             DEPKYD(128)       ,                                    *
                  *     D             SLOPEJ(128)       ,                                    *
                  *     D             SPDKYD(128)                                            *
                  *      COMMON                                                              *
                  *     C / RANGES /                                                         *
                  *     C             NUANMO,                                                *
                  *     C             ANGMAX,                                                *
                  *     C             DELANG,                                                *
                  *     C             DELRAD,                                                *
                  *     D             ANGINT(200)                                            *
                  *     D             RNGMOD(6,200)                                          *
                  *      COMMON                                                              *
                  *     C / RAYPAR /                                                         *
                  *     C             RANGEH,                                                *
                  *     D             BOTLOS(6)         ,                                    *
                  *     D             DRDXDC(6)         ,                                    *
                  *     D             PATHLN(6)         ,                                    *
                  *     D             RANGEC(6)         ,                                    *
                  *     D             SPI(6)            ,                                    *
                  *     D             SPT(6)            ,                                    *
                  *     D             TIR(6)            ,                                    *
                  *     D             TTR    (6)                                             *
                  *      COMMON                                                              *
                  *     C / RAYTRA /                                                         *
                  *     C             NCONCI,                                                *
                  *     C             INITLK,                                                *
                  *     C             Z1     ,                                               *
                  *     C             Z2     ,                                               *
                  *     C             SPVRSO,                                                *
                  *     C             ANGSTR,                                                *
                  *     C             ANGARR,                                                *
                  *     C             ANGBTM,                                                *
                  *     C             ANGSUR,                                                *
                  *     C             SPDVER,                                                *
                  *     C             RANGET                                                 *
                  *C                                                                         *
                  *C      DELRAD      = INCREMENTAL ANGLE                                   (*
                  *C      DEPBOT      = DEPTH OF BOTTOM                                      *
                  *C      NUANMO      = NUMBER OF ANGLES FOR RAYS MINUS ONE           (DIMENS*
                  *C      TIR   ( )   = INITIAL ANGLE OF RAY WITH RANGEH                    (*
                  *C      RANGEH      = HORIZONTAL RANGE BETWEEN SHIPS                       *
                  *C                                                                         *
                  ****************************************************************************
                                                       I
                                                       I
                                                       I
                  ****************************************************************************
I.................*      DO 200                                                              *
I                 *       I             I = 1,6                                              *
I                 ****************************************************************************
I                                                      I
I                                                      I
I                                                      I
I                 ****************************************************************************
I                 *      TIR(I) = 100.0                                                      *
I                 *      TTR(I) = 0.0                                                        *
I                 *      SPI(I) = 0.0                                                        *
I                 *      SPT(I) = 0.0                                                        *
I                 *      RANGEC(I) = 0.0                                                     *
I                 *      PATHLN(I) = 0.0                                                     *
I                 *      DRDXDC(I) = 0.0                                                     *
I                 *      BOTLOS(I) = 0.0                                                     *
I                 ****************************************************************************
I                                                      I
I                                                      I
```



```
      DO 100
     1         J = 1,NUANMO

      IF( RNGMOD(1,J) .EQ. 0.0 )
     * GO TO 100

      J1 = J + 1
      R1 = RNGMOD(1,J1) - RANGEH
      R2 = RNGMOD(1,J) - RANGEH
      R3 = RNGMOD(1,J1) - RNGMOD(1,J)

      IF( R3 .LE. 0.0 )
     * GO TO 90

      IF( (R2 .LE. 0.0) .AND. (R1 .GT. 0.0) )
     * GO TO 10

        GO TO 100

C

   90 CONTINUE

      IF( (R1 .LE. 0.0) .AND. ( R2 .GT. 0.0) )
     * GO TO 10

  100 CONTINUE

        GO TO 200

C

   10 CONTINUE
      TIR(I) = ANGINT(J) - DELRAD*ABS( R2/R3 )
```

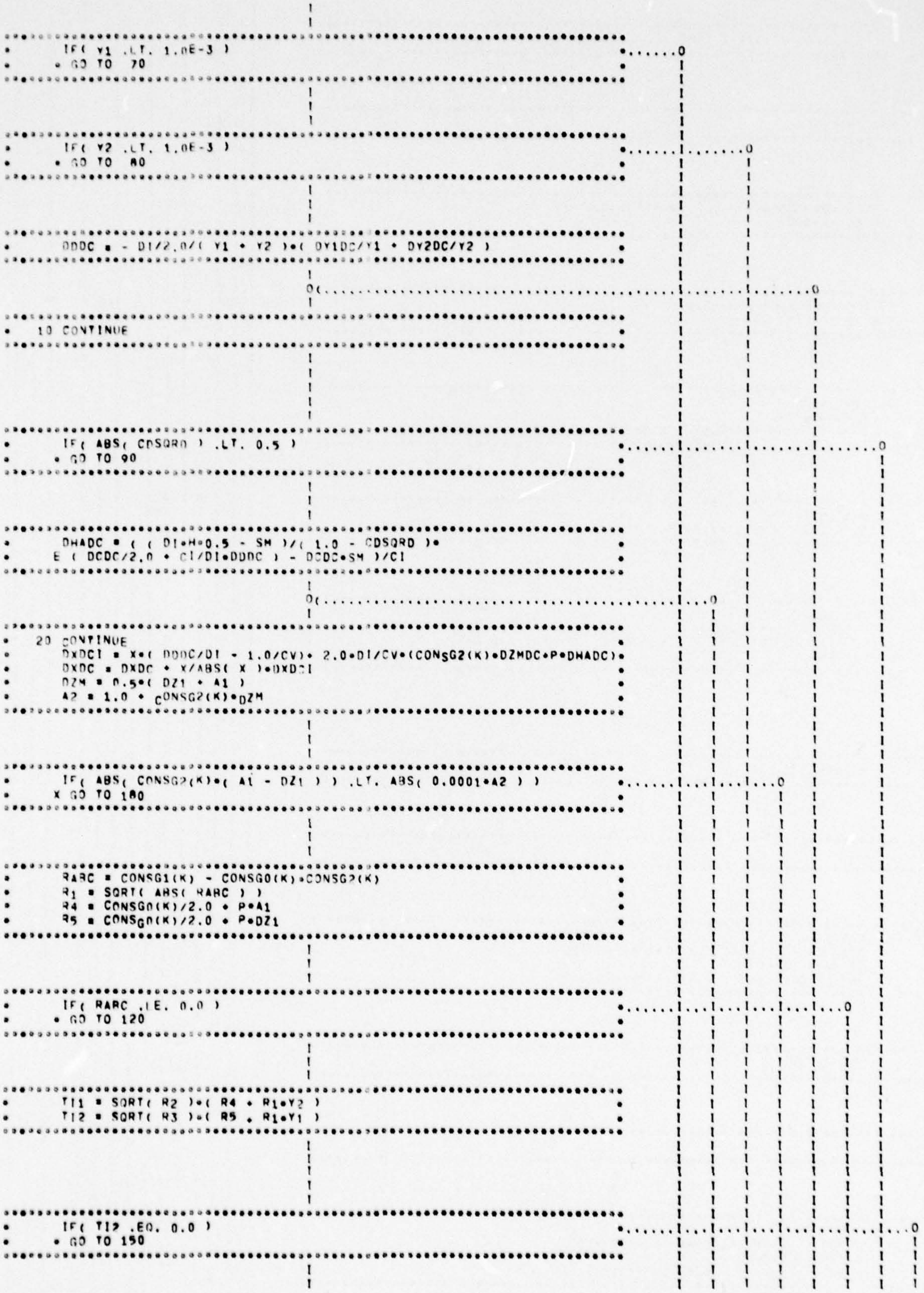


```
  200 CONTINUE

      IF( TIR(1) .EQ. 100.0 )
     * GO TO 30

      CALL
     S           STRIRA
     S ( 1, DEPEVA )

   30 CONTINUE

      IF( TIR(2) .EQ. 100.0 )
     * GO TO 40

      CALL
     S           STRIRA
     S ( 2, 0.0 )
      CALL
     S           LENGTH
     S ( 2, 1, 0.0, DEPEVA )

   40 CONTINUE

      IF( TIR(3) .EQ. 100.0 )
     * GO TO  50

      CALL
     S           STRIRA
     S ( 3, DEPBOT )
      CALL
     S           LENGTH
     S ( 3, NULAPO, 22, DEPEVA )

   50 CONTINUE

      IF( TIR(4) .EQ. 100.0 )
     * GO TO  60

      CALL
     S           STRIRA
     S ( 4, 0.0 )
      CALL
     S           LENGTH
     S ( 4, 1, 0.0, DEPBOT )
      CALL
     S           LENGTH
     S ( 4, NULAPO, 22, DEPEVA )
```

```
                                I
                                I
******************************************************************
*   60 CONTINUE                                                  *
******************************************************************
                                I
                                I
                                I
******************************************************************
*       IF( TIR(5) .EQ. 100,0 )                                  *......O
*     * GO TO  70                                                *      I
******************************************************************      I
                                I                                       I
                                I                                       I
                                I                                       I
******************************************************************      I
*       IF( TIR(5) .LT. 0,0 )                                    *......I......O
*     * GO TO 80                                                 *      I      I
******************************************************************      I      I
                                I                                       I      I
                                I                                       I      I
                                I                                       I      I
******************************************************************      I      I
*       CALL                                                     *      I      I
*      S            STRTRA                                       *      I      I
*      S ( 5, DEPBOT )                                           *      I      I
*       CALL                                                     *      I      I
*      S            LENGTH                                       *      I      I
*      S ( 5, NULAPO, 72, 0,0 )                                  *      I      I
******************************************************************      I      I
                                I                                       I      I
                                I                                       I      I
                                I                                       I      I
******************************************************************      I      I
*       IF( ANGSTR .LT. 0,0 )                                    *......I.....)V
*     * GO TO 80                                                 *      I      I
******************************************************************      I      I
                                I                                       I      I
                                I                                       I      I
                                I                                       I      I
******************************************************************      I      I
*       CALL                                                     *      I      I
*      S            LENGTH                                       *      I      I
*      S ( 5, 1, 0,0, DEPEVA )                                   *      I      I
******************************************************************      I      I
                                I                                       I      I
                                I                                       I      I
                                I                                       I      I
******************************************************************      I      I
*        GO TO 70                                                *.....)V      I
******************************************************************      I      I
                                                                        I      I
                                O(......................................I......O
                                I                                       I
******************************************************************      I
*   80 CONTINUE                                                  *      I
*      WRITE ( 6, 1000 )                                         *      I
*     W TIR(5)                                                   *      I
* 1000 FORMAT(                                                   *      I
*     F 38H WRONG STARTING ANGLE FOR RAY TYPE 5 =  E20.10        *      I
*     F )                                                        *      I
******************************************************************      I
                                I                                       I
                                I                                       I
                                I                                       I
                                O(......................................O
                                I
******************************************************************
*   70 CONTINUE                                                  *
******************************************************************
                                I
                                I
                                I
******************************************************************
*       IF( TIR(6) .EQ. 100,0 )                                  *......O
*     * GO TO 900                                                *      I
******************************************************************      I
                                I                                       I
                                I                                       I
```

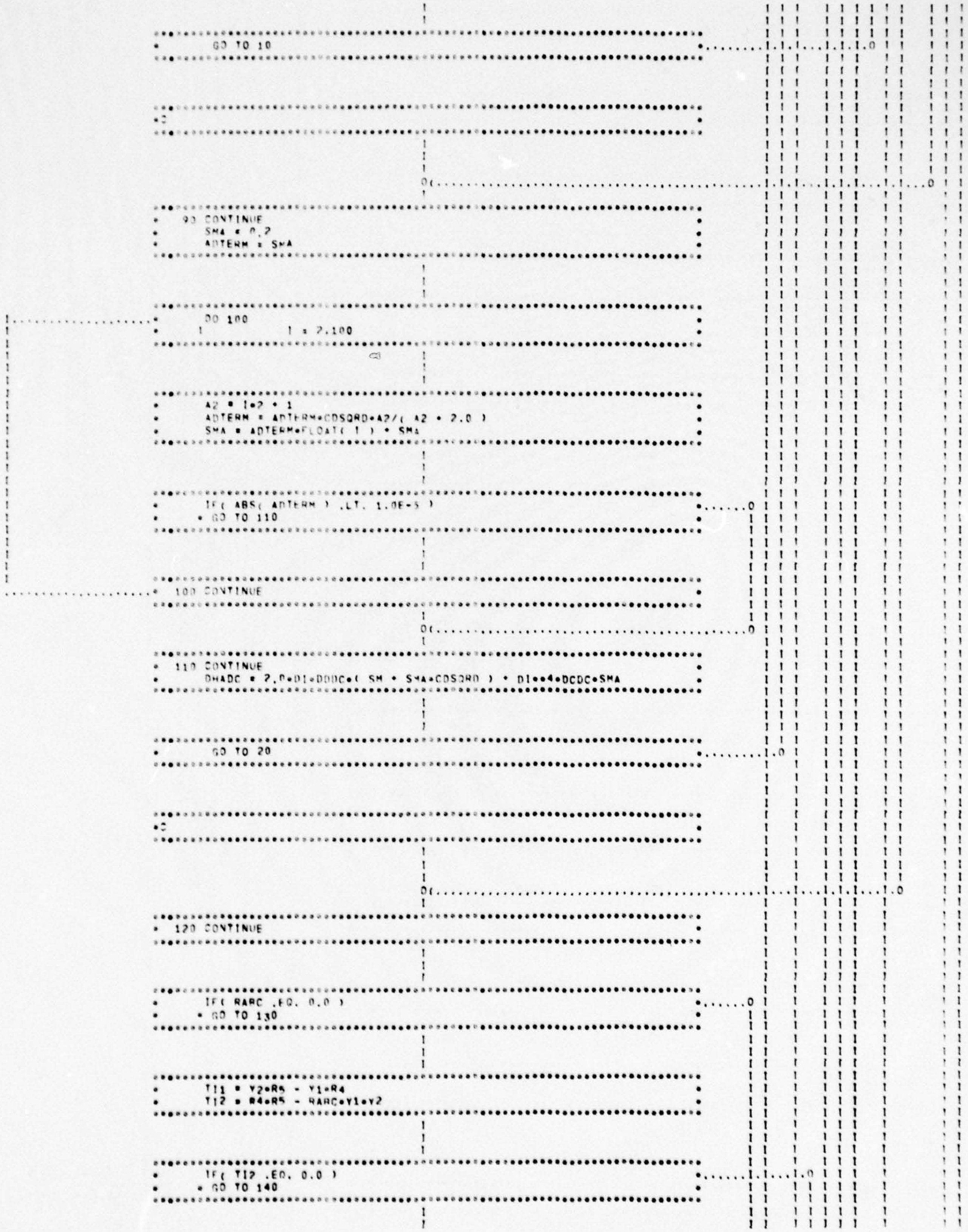
```
                                   I                                        I
**********************************************************************      I
*      CALL                                                          *      I
*     S          STRTRA                                              *      I
*     S ( 6, 0.0 )                                                   *      I
*      CALL                                                          *      I
*     S          LENGTH                                              *      I
*     S ( 6, 1, 0.0, DEPROT )                                        *      I
*      CALL                                                          *      I
*     S          LENGTH                                              *      I
*     S ( 6, NULAPO, 72, 0.0 )                                       *      I
*      CALL                                                          *      I
*     S          LENGTH                                              *      I
*     S ( 6, 1, 0.0, DEPEVA )                                        *      I
**********************************************************************      I
                                   I                                        I
                                   0(.......................................0
                                   I
**********************************************************************
*  900 CONTINUE                                                      *
**********************************************************************
                                   I
                                   I
                                   I
**********************************************************************
*      RETURN                                                        *
**********************************************************************


**********************************************************************
*C                                                                   *
**********************************************************************
                                   I
                                   I
                                   I
**********************************************************************
*      END                                                           *
**********************************************************************
```

```
                              (ENTRANCE)
                                  I
                                  I
*******************************************************************************
*CRAYPTH00          SUBROUTINE RAYPTH                                         *
*CSPATH000          SUBROUTINE CONTROLLED BY NCONCI                           *
*       SUBROUTINE                                                            *
*      S            RAYPTH                                                    *
*C                                                                            *
*C      ***********************************************************************
*C      *                                                                    **
*C      THIS COMPUTES REQUIRED RAY PARAMETERS SUCH AS TIME AND PATH LENGTH*
*C      *                                                                    **
*C      ***********************************************************************
*C                                                                            *
*       COMMON                                                                *
*      C / LCONST /                                                           *
*      C            NULAPO,                                                   *
*      C            DEPHOT,                                                   *
*      D            CONSG0(128)      ,                                        *
*      D            CONSG1(128)      ,                                        *
*      D            CONSG2(128)      ,                                        *
*      D            CONSV0(128)      ,                                        *
*      D            DELTAZ(128)      ,                                        *
*      D            DEPKYD(128)      ,                                        *
*      D            SLOPEJ(128)      ,                                        *
*      D            SPDKYD(128)                                               *
*       COMMON                                                                *
*      C / PTHLNG /                                                           *
*      C            AI     ,                                                  *
*      C            BI     ,                                                  *
*      C            COSQRO,                                                   *
*      C            CI     ,                                                  *
*      C            DI     ,                                                  *
*      C            DXDC   ,                                                  *
*      C            DZ1    ,                                                  *
*      C            DZM    ,                                                  *
*      C            K      ,                                                  *
*      C            PL     ,                                                  *
*      C            SM     ,                                                  *
*      C            H      ,                                                  *
*      C            X      ,                                                  *
*      C            Y1     ,                                                  *
*      C            Y2     ,                                                  *
*      C            TIMCON                                                    *
*       COMMON                                                                *
*      C / RAYTRA /                                                           *
*      C            NCONCI,                                                   *
*      C            INITLK,                                                   *
*      C            Z1     ,                                                  *
*      C            Z2     ,                                                  *
*      C            SPVRSQ,                                                   *
*      C            ANGSTR,                                                   *
*      C            ANGARR,                                                   *
*      C            ANGRTM,                                                   *
*      C            ANGSUR,                                                   *
*      C            CV     ,                                                  *
*      C            RANGET                                                    *
*C                                                                            *
*C      A1          = DEPTH FROM  END  OF RAY PORTION TO TOP OF LAYER         *
*C                  ALWAYS POSITIVE OR ZERO                                   *
*C      DXDC        = RANGE DERIVATIVE                                        *
*C      DZ1         = DEPTH FROM START OF RAY PORTION TO TOP OF LAYER         *
*C                  ALWAYS POSITIVE OR ZERO                                   *
*C      TIMCON      = PROPAGATION TIME                                        *
*C      PL          = PATH LENGTH (IN K-YD) OF RAY                            *
*C      CV          = VERTEX VELOCITY (IN K-YD/SEC)                           *
*C      Y1          =                                                         *
*C      Y2          =                                                         *
*C      TI          = TIME INCREMENT FOR RAY TO GO X K-YD                     *
*C                                                                            *
*       NAMELIST                                                              *
*      N / TIMES                                                              *
*      N TOMUCH,                                                              *
*      NA1,A2,A3,ADTERM, AI, BI, COSQRO, CI, DADC, DCDC, DDDC, DHADC, DI,    *
*      N DXDC, DXDCI, D[illegible], DY2DC, DZ1, DZ1DC, DZ2DC, DZMDC, H, K, P, PL,*
*      N QI, R1, R2, R3, [illegible], R5, RABC, SM, SMA, SPDVER, SPVRSQ, TI, TI1,  *
*      N TI2, TIMCON, TJ, TOMUCH, X, X1, X2, Y1, Y2                          *
*C                                                                            *
*       IF( NCONCI .NE. 2 )                                                   *
*      * RETURN                                                               *
*       P = CONSG1(K) - 0.5*CONSG0(K)*CONSG2(K)                               *
*       DADC = 2.0/CV*SPVRSQ                                                  *
*       DCDC = CONSG2(K)**2*DADC                                              *
*       R2 = (1.0 + CONSG2(K)*DZ1)**2                                         *
*       DY1DC = R2*DADC                                                       *
*       R3 = ( 1.0 + CONSG2(K)*A1 )**2                                        *
*       DY2DC = R3*DADC                                                       *
*       DZMDC = 0.0                                                           *
*******************************************************************************
                                  .
```

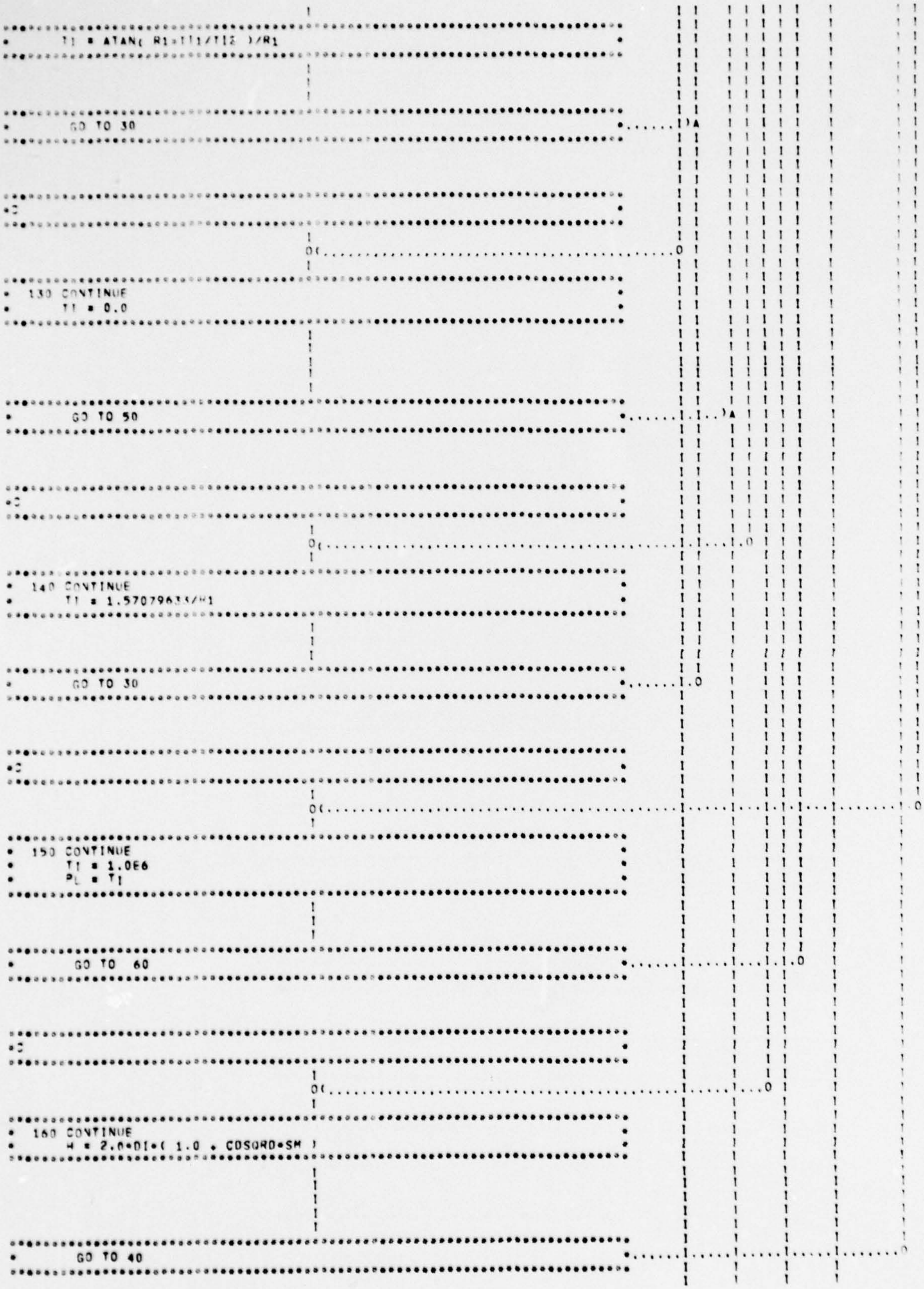


```
      IF( Y1 .LT. 1.0E-3 )
     * GO TO  70

      IF( Y2 .LT. 1.0E-3 )
     * GO TO  80

      DDDC = - DI/2.0/( Y1 + Y2 )*( DY1DC/Y1 + DY2DC/Y2 )

   10 CONTINUE

      IF( ABS( CDSQRD ) .LT. 0.5 )
     * GO TO 90

      DHADC = ( ( DI*H*0.5 - SM )/( 1.0 - CDSQRD )*
    E ( DCDC/2.0 + CI/DI*DDDC ) - DCDC*SM )/CI

   20 CONTINUE
      DXDCI = X*( DDDC/DI - 1.0/CV)+ 2.0*DI/CV*(CONSG2(K)*DZMDC+P*DHADC)
      DXDC = DXDC + X/ABS( X )*DXDCI
      DZM = 0.5*( DZ1 + A1 )
      A2 = 1.0 + CONSG2(K)*DZM

      IF( ABS( CONSG2(K)*( A1 - DZ1 ) ) .LT. ABS( 0.0001*A2 ) )
     X GO TO 180

      RABC = CONSG1(K) - CONSG0(K)*CONSG2(K)
      R1 = SQRT( ABS( RABC ) )
      R4 = CONSG0(K)/2.0 + P*A1
      R5 = CONSG0(K)/2.0 + P*DZ1

      IF( RABC .LE. 0.0 )
     * GO TO 120

      TI1 = SQRT( R2 )*( R4 + R1*Y2 )
      TI2 = SQRT( R3 )*( R5 + R1*Y1 )

      IF( TI2 .EQ. 0.0 )
     * GO TO 150
```

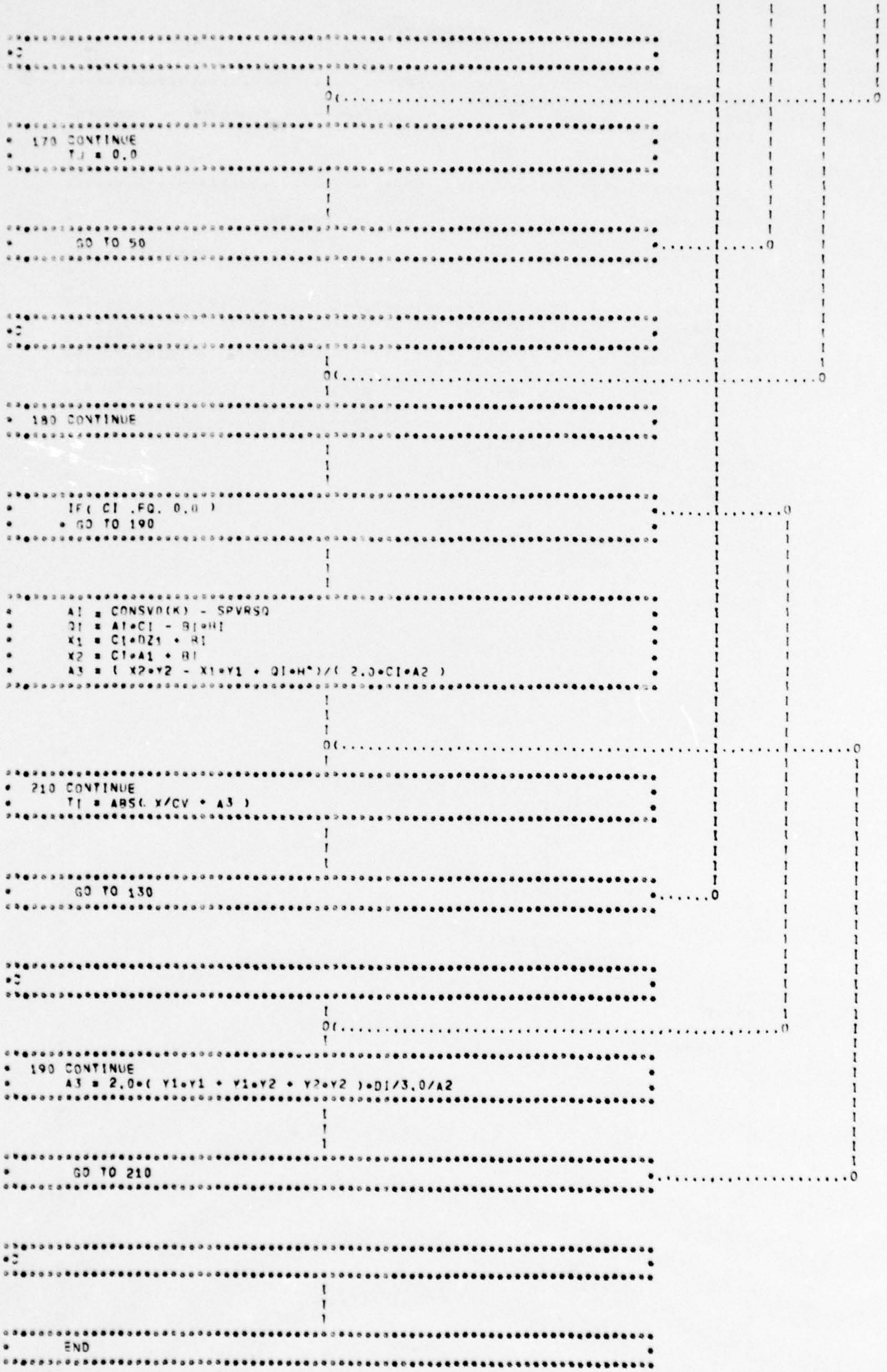
```
                                       I
***************************************************************
*      TI = ALOG( TI1/TI2 )/R1                                *
***************************************************************
                                       I
                                       O(..................................O
                                       I
***************************************************************
*   30 CONTINUE                                               *
***************************************************************
                                       I
                                       I
                                       I
***************************************************************
*      IF( ABS( CDSQRD ) .LT. 0.5 )                           *.............O
*     * GO TO 160                                             *
***************************************************************
                                       I
                                       I
                                       I
***************************************************************
*      IF( CONSG2(K) .EQ. 0.0 )                               *..................O
*     * GO TO 170                                             *
***************************************************************
                                       I
                                       O(.......................................................O
                                       I
***************************************************************
*   40 CONTINUE                                               *
*      TOMUCH = TI                                            *
*      TJ = ( CONSG1(K)*CV*X - 2.0*P*H + RABC*TI )/CONSG2(K)**2 *
***************************************************************
                                       I
                                       O(..........................................O
                                       I
***************************************************************
*   50 CONTINUE                                               *
*      TI = ABS( CONSV0(K)*CV*X + TJ )                        *
***************************************************************
                                       I
                                       O(...............................................O
                                       I
***************************************************************
*   60 CONTINUE                                               *
*      TIMCON = TI + TIMCON                                   *
*      PL = ( FVELOC( D/1, K ) + FVELOC( A1, K ) )/2.0*TI + PL *
*  900 CONTINUE                                               *
***************************************************************
                                       I
                                       I
                                       I
                                       I
                                       I
***************************************************************
*      RETURN                                                 *
***************************************************************


***************************************************************
*C                                                            *
***************************************************************
                                       I
                                       O(..................................O
                                       I
***************************************************************
*   70 CONTINUE                                               *
*      DZ1DC = -0.5*R2/( CI*DZ1 + BI )*DADC                   *
*      DZMDC = DZ1DC/2.0                                      *
*      DDDC = DZ1DC/Y2 - DI/Y2/Y2*DY2DC/2.0                   *
***************************************************************
                                       I
                                       I
                                       I
***************************************************************
*       GO TO 10                                              *...................A


***************************************************************
*C                                                            *
***************************************************************
                                       I
                                       O(........................................O
                                       I
***************************************************************
*   80 CONTINUE                                               *
*      DZ2DC = -0.5*R3/( CI*A1 + BI )*DADC                    *
*      DZMDC = DZ2DC/2.0                                      *
*      DDDC = DZ2DC/Y1 - 0.5/Y1*DI/Y1*DY1DC                   *
***************************************************************
                                       I
```



```
      GO TO 10

C

   90 CONTINUE
      SMA = 0.2
      ADTERM = SMA

      DO 100
    1            I = 2,100

      A2 = I*2 + 1
      ADTERM = ADTERM*CDSQRD*A2/( A2 + 2.0 )
      SMA = ADTERM*FLOAT( I ) + SMA

      IF( ABS( ADTERM ) .LT. 1.0E-5 )
     * GO TO 110

  100 CONTINUE

  110 CONTINUE
      DHADC = 2.0*DI*DDDC*( SM + SMA*CDSQRD ) + DI**4*DCDC*SMA

      GO TO 20

C

  120 CONTINUE

      IF( RARC .EQ. 0.0 )
     * GO TO 130

      TI1 = Y2*R5 - Y1*R4
      TI2 = R4*R5 - RARC*Y1*Y2

      IF( TI2 .EQ. 0.0 )
     * GO TO 140
```



```
      TI = ATAN( R1*TI1/TI2 )/R1
      GO TO 30
C
  130 CONTINUE
      TI = 0.0
      GO TO 50
C
  140 CONTINUE
      TI = 1.57079633/R1
      GO TO 30
C
  150 CONTINUE
      TI = 1.0E6
      PL = TI
      GO TO  60
C
  160 CONTINUE
      H = 2.0*DI*( 1.0 + COSQRD*SM )
      GO TO 40
```

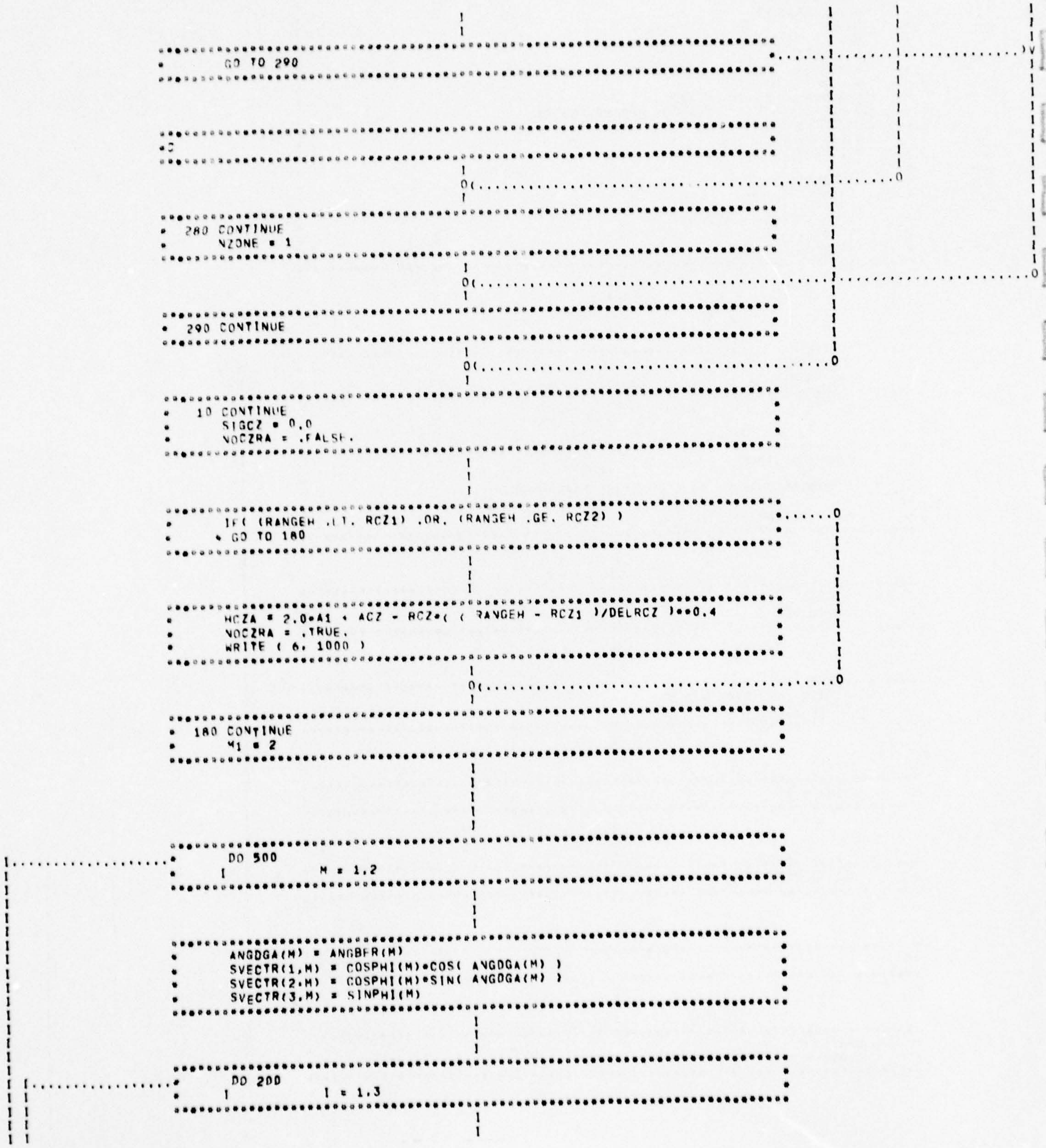


```
C

  170 CONTINUE
      TI = 0.0

      GO TO 50

C

  180 CONTINUE

      IF( CI .EQ. 0.0 )
     * GO TO 190

      AI = CONSVO(K) - SPVRSQ
      QI = AI*CI - BI*HI
      X1 = CI*DZ1 + BI
      X2 = CI*A1 + BI
      A3 = ( X2*Y2 - X1*Y1 + QI*H* )/( 2.0*CI*A2 )

  210 CONTINUE
      TI = ABS( X/CV + A3 )

      GO TO 130

C

  190 CONTINUE
      A3 = 2.0*( Y1*Y1 + Y1*Y2 + Y2*Y2 )*DI/3.0/A2

      GO TO 210

C

      END
```

```
                                (ENTRANCE)
                                    I
                                    I
*********************************************************************
*CRECEIVOO      SUBROUTINE RECEIV                                   *
*CIGNALDOO      THIS COMPUTES SIGNALS RECIEVED AS A FUNCTION OF FREQUEN*
*    SUBROUTINE                                                     *
*    S          RECIEV                                              *
*C                                                                  *
*C   ***************************************************************
*C   *                                                             **
*C   THIS COMPUTES REQUIRED NOISE AND SIGNAL SPECTRAE               *
*C   *                                                             **
*C   ***************************************************************
*C                                                                  *
*C                                                                  *
*     COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*    1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*    2NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*    3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*    4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*    5LSUBE,ALSUBS,SINPSE,SINPEV,MSCO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*    6,FOS,FBWS,FIS,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBSE,F1E,F2E*
*    7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*    8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                  *
*     COMMON / ANGRAD / ANGBER(2)                                   *
*     COMMON                                                        *
*    C / ARRAYP /                                                   *
*    D          ANGDGA(2)       .                                   *
*    D          ARRAYD(3,2)     .                                   *
*    D          COSPHI(2)       .                                   *
*    D          COSRAD(2)       .                                   *
*    C          NSHIPS,                                             *
*    D          SINPHI(2)       .                                   *
*    D          SINRAD(2)       .                                   *
*    D          TARRIV(3,2)     .                                   *
*    D          TMATRX(3,3,2)   .                                   *
*    D          TS01   (3)      .                                   *
*    D          TS02   (3)      .                                   *
*    DDUMMYS                                                        *
*     COMMON                                                        *
*    C / FREQUN /                                                   *
*    D          COSAVE(2)       .                                   *
*    C          DELRCZ,                                             *
*    D          DYHITH(2)       .                                   *
*    D          ARAREA(2)       .                                   *
*    C          CTWOPI,                                             *
*    D          SDAREA(2)       .                                   *
*    D          FRQSIG(128,2)   .                                   *
*    D          FRQLCN(128,2)   .                                   *
*    D          FRQIDB(128,2)   .                                   *
*    D          ANGDEP(2)       .                                   *
*    D          FRQTRN(128,2)   .                                   *
*    D          FRQASD(128,2)   .                                   *
*    D          FRQNCN(128,2)   .                                   *
*    D          FRQNOS(128,2)   .                                   *
*    D          NUMFRQ(2)       .                                   *
*    DDUM9                                                          *
*     COMMON                                                        *
*    C / ARRAYC /                                                   *
*    D          NARRAY(2)       .                                   *
*    C          ARRYH1,                                             *
*    C          ARRYH2,                                             *
*    D          ARWIDT(2),                                          *
*    C          DELF  ,                                             *
*    D          FRQRES(2)       .                                   *
*    C          QTRAN1,                                             *
*    C          QTRAN2,                                             *
*    NDUM4                                                          *
*     COMMON                                                        *
*    C / CONSTN /                                                   *
*    C          DEPSER,                                             *
*    C          NCONSK,                                             *
*    C          SPATSE,                                             *
*    C          DEPEVA,                                             *
*    C          NCONSL,                                             *
*    C          SPATEV                                              *
*      COMMON
*     C / RAYPAR /
*     C           RANGEH,
*     D           BOTLOS(6)        ,
*     D           DRDXDC(6)        ,
*     D           PATHLN(6)        ,
*     D           RANGEC(6)        ,
*     D           SPRLOS(6,2)      ,
*     D           ANGTER(6,2)
*      COMMON
*     C / INDEXS /
*     D           ANDISO(16,2)    ,
*     D           DIRSON(50,16,2)  ,
*     D           DLDIAN(16,2)     ,
*     D           NUDIAN(2)        ,
*     CDUMMYA
*      COMMON
*     C / RCONST /
*     C           ACZ    ,
*     C           AMLSRD,
*     C           BCZ    ,
*     C           HCI    ,
*     C           HZSD   ,
*     C           NCONSD,
*     C           RCZ1   ,
*     C           RCZ2   ,
*     C           SDCON  ,
*     D           ANCZAV(2)        ,
*     C           ZW     ,
*     NDUM5
*      COMMON
*     C / SURDUC /
*     C           BLA1   ,
*     C           BLA2   ,
*     C           BLA3   ,
*     C           DTRAD  ,
*     C           J      ,
*     C           M      ,
*     D           BAFFUN(128,2)    ,
*     D           DELBAF(128,2)    ,
*     D           FLONOS(128,2)   ,
*     D           RADSPC(40,50,2)  ,
*     C           NTIMEN
*      COMMON
*     C / SURFAC /
*     C           A6     ,
*     C           CONST2,
*     C           CONST4,
*     C           CZANGL,
*     C           CZANDL,
*     C           CZRANG,
*     C           G1SD   ,
*     C           G2SD   ,
*.    C           NCZRAS,
*     C           NZONE  ,
*     C           RSD    ,
*     C           RSD1   ,
*     C           SCSD   ,
*     C           SQRTZL,
*     C           SS     ,
*     C           ZL     ,
*     CDUMMYZ
*C
*      DIMENSION
*     D           CONPHI(6)        ,
*     D           CONSBF(6)        ,
*     D           NCNABT(6)        ,
*     D           POWSIG(128,2)    ,
*     D           POWNOS(128,2)    ,
*     D           SVECTR(3,2)      ,
*     D           SO1    (3)       ,
*     D           SO2    (3)       ,
*     D           TVECTR(3,2)      ,
*     DDUMMYS(1)
*C
*      LOGICAL
*     L NOCZRA,
*     L NOSDRA
*C
```

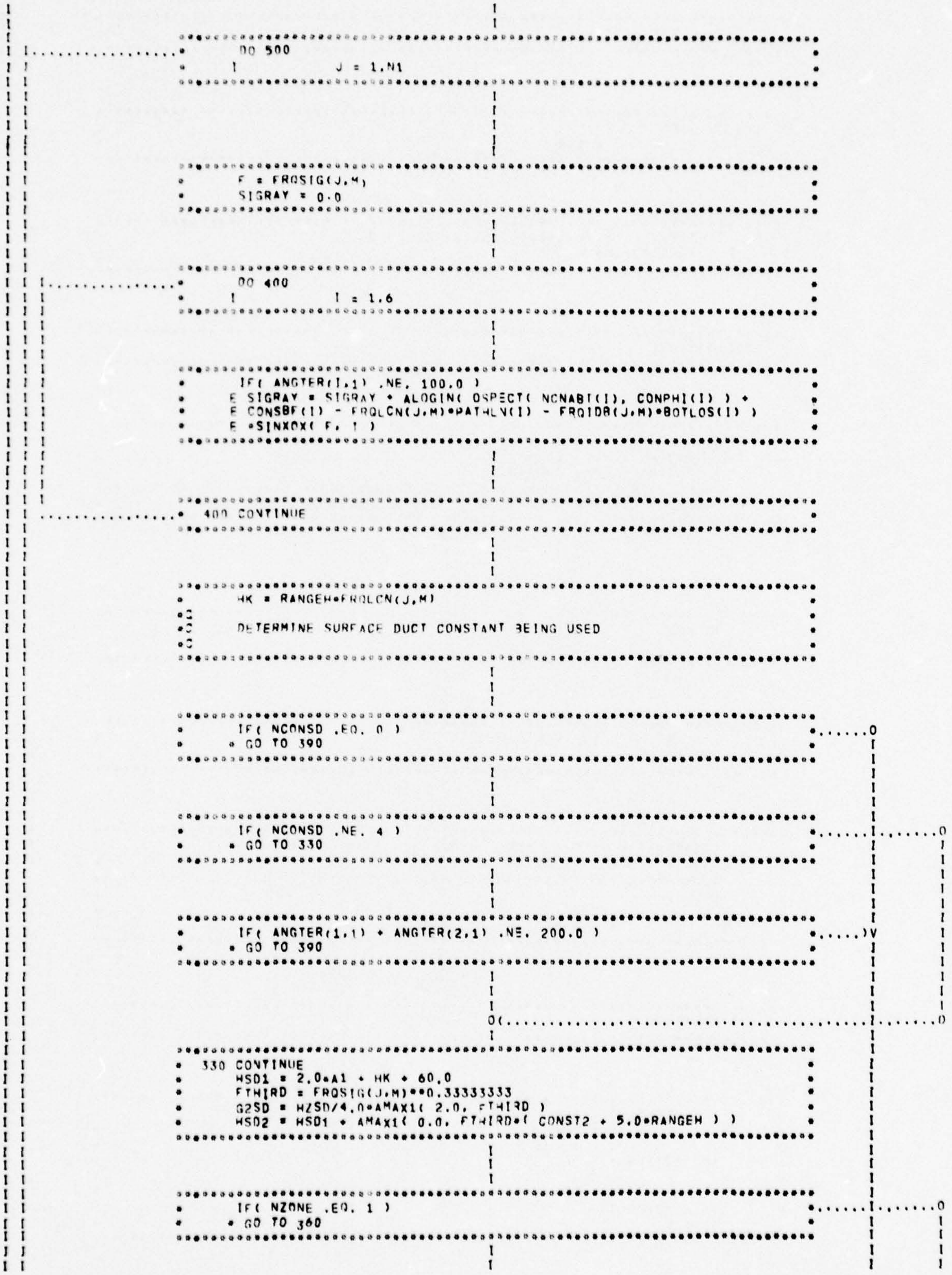
```
*      EQUIVALENCE                                                         *
*     Q ( SVECTR, SO1 ),                                                   *
*     Q ( SVECTR(1,2), SO2 ),                                              *
*     Q ( TVECTR, TSO1 )                                                   *
*C                                                                         *
*C     ABEAM1      = AZIMUTHAL ANGLE FOR SEARCHER STEERING                (*
*C     ABEAM2      = AZIMUTHAL ANGLE FOR EVADER STEERING                  (*
*C     ACZ         = CONVERGENT ZONE CONSTANT                              *
*C     ANCZAV( )   = AVERAGE CONVERGENT ZONE ANGLES                       (*
*C     ANDISO( , )= ANGLES OF DIFFERENT DIRECTIVITIES FOR SHIPS           (*
*C     ANGAZM( )   = AZIMUTHAL STEERING ANGLES FOR EACH SHIP              (*
*C     ANGBER( )   = RELATIVE BEARING ANGLE FOR EACH SHIP                 (*
*C     ARAD1       = AZIMUTHAL ANGLE OF SEARCHER MEASURED FROM EVADER      *
*C     ARAD2       = AZIMUTHAL ANGLE OF EVADER MEASURED FROM SEARCHER      *
*C     ARAREA( )   = ARRAY AREA AS A FUNCTION OF SHIP                      *
*C     ARRAY2      = TYPE OF ARRAY CONTROL CONSTANT FOR EVADER     (DIMENS*
*C     ARRAY1      = TYPE OF ARRAY CONTROL CONSTANT FOR SEARCHER   (DIMENS*
*C     ARRAYD( , )= MODIFIED ARRAY DIMENSIONS FOR SHIPS                    *
*C     ARRYH1      = ARRAY HEIGTH ON SEARCHER                              *
*C     ARRYH2      = ARRAY HEIGTH ON EVADER                                *
*C     ARRAYW1     = ARRAY WIDTH ON SEARCHER                               *
*C     ARRAYW2     = ARRAY WIDTH ON EVADER                                 *
*C     ARWIDT( )   = WIDTH OF ARRAY AS A FUNCTION OF SHIP                  *
*C     ASNXOX      = FUNCTION TO COMPUTE SIN(X)/X                          *
*C     BANGLE      = ABSOLUTE VALUE OF BOTTOM ANGLE                       (*
*C     BCZ         = CONVERGENT ZONE CONSTANT                              *
*C     BF1    ( )  = BAFFLING CORRECTION FACTOR FOR SEARCHER               *
*C                 AS A FUNCTION OF ANGLE                                  *
*C     BF2    ( )  = BAFFLING CORRECTION FACTOR FOR EVADER                 *
*C                 AS A FUNCTION OF ANGLE                                  *
*C     BFOFCZ( )   = BAFFLING CONSTANT FOR CONVERGENT ZONE         (DIMENS*
*C     BLA1        = BOTTOM LOSS CONSTANT                                  *
*C     BLA2        = BOTTOM LOSS CONSTANT                                  *
*C     BLA3        = BOTTOM LOSS CONSTANT                                  *
*C     CDYOVV( )   = EFFECTIVE ARRAY HEIGTH AND SPEED CONSTANT             *
*C     CONFBC      = FUNCTION TO COMPUTE CONSTANT FOR BC CALCULATIONS      *
*C     COSAVE( )   = COSINE OF AVERAGE CONVERGENT ZONE ANGLES      (DIMENS*
*C     DBEAM1      = DEPRESSION ANGLE FOR SEARCHER STEERING               (*
*C     DBEAM2      = DEPRESSION ANGLE FOR EVADER STEERING                 (*
*C     DIRSON(,,) = DIRECTIVITY INDEX FOR SONAR                            *
*C     DLDIAN( , )= DIFFERENCES IN ANGLES OF DIRECTIVITIES                (*
*C     DSIG        = DEPRESSION ANGLE OF RAY                              (*
*C     DTRAD       = ANGLE INCREMENT FOR RADIATED SIGNAL                  (*
*C     DYHITH( )   = HEIGTH CONSTANT FOR ARRAY SURFACE DUCT        (DIMENSI*
*C     FLN1   ( )  = PLANE WAVE FLOW NOISE FOR SEARCHER                    *
*C     FLN2   ( )  = PLANE WAVE FLOW NOISE FOR EVADER                      *
*C     FLONOS( , )= PLANE WAVE FLOW NOISE AS A FUNCTION OF FREQUENCY       *
*C     FRAD11      = LOWER FREQUENCY LIMIT OF SEARCHER EQUIPMENT           *
*C     FRAD12      = UPPER FREQUENCY LIMIT OF SEARCHER EQUIPMENT           *
*C     FRAD21      = LOWER FREQUENCY LIMIT OF EVADER EQUIPMENT             *
*C     FRAD22      = UPPER FREQUENCY LIMIT OF EVADER EQUIPMENT             *
*C     FRES1       = TRANSDUCER RESONANT FREQUENCY ON SEARCHER             *
*C     FRES2       = TRANSDUCER RESONANT FREQUENCY ON EVADER               *
*C     HZSD        = SURFACE DUCT CONSTANT                                 *
*C     NARRAY( )   = CONTROL PARAMETER DEFINING TYPE OF ARRAY      (DIMENS*
*C     NCNACZ( )   = BAFFLING ANGLE NUMBER AS A FUNCTION OF SHIP   (DIMENS*
*C     NUDIAN( )   = NUMBER OF DIRECTIVITY ANGLES FOR SHIPS        (DIMENS*
*C     NUFREV      = NUMBER OF FREQUENCY POINTS FOR EVADER         (DIMENS*
*C     NUFRSE      = NUMBER OF FREQUENCY POINTS FOR SEARCHER       (DIMENS*
*C     POR         = POROSITY OF BOTTOM                                    *
*C     POWNOS( , )= NOISE POWER SPECTRUM AS A FUNCTION OF FREQUENCY        *
*C     POWSIG( , )= SIGNAL POWER SPECTRUM AS A FUNCTION OF FREQUENCY       *
*C     QTRAN1      = TRANSDUCER FIGURE OF MERIT ON SEARCHER                *
*C     QTRAN2      = TRANSDUCER FIGURE OF MERIT ON EVADER                  *
*C     SDCON       = WAVE HEIGTH PARAMETER                                 *
*C     SDCON       = SURFACE DUCT CONSTANT                                 *
*C     SSD         =                                                       *
*C     TCZAV1      =              AVERAGE OF CONVERGENT ZONE ANGLES       (*
*C     TCZAV2      =              AVERAGE OF CONVERGENT ZONE ANGLES       (*
*C     V           = PROPAGATION SPEED AT END POINT OF RAY               (K*
*C     XSD         = SEE -AMLSRD-                                      (SQR*
*C     XSD         = SURFACE DUCT CONSTANT                                 *
*C     ZW          = HEIGTH OF SURFACE WAVES                               *
*C                                                                         *
```

```
*     NAMELIST                                           *
*    N / LOSSES /                                        *
*    N MSHIPS,                                           *
*    N    F, SIGRAY, SIGCZ, SIGSD, HK                    *
*    N . SETSUP /                                        *
*    N NOSDRA, NOCZRA, ANGHER, R1, A1, HCZA, HCY         *
*    N / SHIPCO /                                        *
*    N NCNASD, NCNACZ, NCNABT,                           *
*    N CONASD, CONACZ, BFOFSD, BFOFCZ, CONSBF            *
*    N / STATUS /                                        *
*    N N1, K, M, AFRAC                                   *
*    N , NDUD, DUD, DUM, DI                              *
*C                                                       *
*     ANGBER(1) = BSP                                    *
*     ANGRER(2) = BEP                                    *
*     COSRAD(2) = COS( BEP )                             *
*     A1 = TNLG10( RANGEH )                              *
*     SIGSD = 0.0                                        *
*     NOSDRA = .FALSE.                                   *
**********************************************************
                            I
                            I
                            I
                            I
                            I
**********************************************************
*     IF( NCONSD .EQ. 0 )                                *.....O
*      * GO TO 10                                        *     I
**********************************************************     I
                            I                                  I
                            I                                  I
                            I                                  I
**********************************************************     I
*     NOSDRA = .TRUE.                                    *     I
*C                                                       *     I
*C    DETERMINE SURFACE ZONE CONSTANT PARAMETER          *     I
*C                                                       *     I
*     RSD = RANGEH/SORIZL                                *     I
**********************************************************     I
                            I                                  I
                            I                                  I
**********************************************************     I
*     IF( RSD .LT. RSD1 )                                *.....I.....O
*      * GO TO 280                                       *     I     I
**********************************************************     I     I
                            I                                  I     I
                            I                                  I     I
                            I                                  I     I
**********************************************************     I     I
*     IF( RSD .GE. RSD1 + 0.5 )                          *.....I.....I.....O
*      * GO TO 270                                       *     I     I     I
**********************************************************     I     I     I
                            I                                  I     I     I
                            I                                  I     I     I
**********************************************************     I     I     I
*     NZONE = 2                                          *     I     I     I
**********************************************************     I     I     I
                            I                                  I     I     I
                            I                                  I     I     I
**********************************************************     I     I     I
*       GO TO 290                                        *.....I.....I.....I.....O
**********************************************************     I     I     I     I
                                                               I     I     I     I
                                                               I     I     I     I
**********************************************************     I     I     I     I
*C                                                       *     I     I     I     I
**********************************************************     I     I     I     I
                            I                                  I     I     I     I
                            O(.................................I.....I.....O     I
                                                               I     I           I
**********************************************************     I     I           I
*  270 CONTINUE                                          *     I     I           I
*     NZONE = 3                                          *     I     I           I
**********************************************************     I     I           I
                            I                                  I     I           I
                            I                                  I     I           I
```



```
      GO TO 290

C

  280 CONTINUE
      NZONE = 1

  290 CONTINUE

   10 CONTINUE
      SIGCZ = 0.0
      NOCZRA = .FALSE.

      IF( (RANGEH .LT. RCZ1) .OR. (RANGEH .GE. RCZ2) )
     + GO TO 180

      HCZA = 2.0*A1 + ACZ - RCZ*( ( RANGEH - RCZ1 )/DELRCZ )**0.4
      NOCZRA = .TRUE.
      WRITE ( 6, 1000 )

  180 CONTINUE
      M1 = 2

      DO 500
      I          M = 1.2

      ANGDGA(M) = ANGBER(M)
      SVECTR(1,M) = COSPHI(M)*COS( ANGDGA(M) )
      SVECTR(2.M) = COSPHI(M)*SIN( ANGDGA(M) )
      SVECTR(3,M) = SINPHI(M)

      DO 200
      I          I = 1.3
```

```
I I                                        I
I I       ***********************************************************************
I I       *     TVECTR(I,M) = 0.0                                               *
I I       ***********************************************************************
I I                                        I
I I                                        I
I I                                        I
I I       ***********************************************************************
I I I.....*     DO 200                                                          *
I I I     *     I          J = 1,3                                              *
I I I     ***********************************************************************
I I I                                      I
I I I                                      I
I I I                                      I
I I I     ***********************************************************************
I I I     *     TVECTR(I,M) = TMATRX(I,J,M)*SVECTR(J,M)                         *
I I I     *    E + TVECTR(I,M)                                                  *
I I I     ***********************************************************************
I I I                                      I
I I I                                      I
I I I                                      I
I I I     ***********************************************************************
I   ......*  200 CONTINUE                                                       *
I         ***********************************************************************
I                                          I
I                                          I
I                                          I
I         ***********************************************************************
I         *     MSHIPS = M                                                      *
I         *     SINRAD(M) = SIN( ANGBER(M) )                                    *
I         *     COSRAD(M) = COS( ANGBER(M) )                                    *
I         *     BFOFCZ = BAFFLE( NCNACZ, ACOS( COSRAD(M)*COSAVE(M) ), CONACZ)-HCZA*
I         *     BFOFSD = BAFFLE( NCNASD, ANGBER(M), CONASD )                    *
I         ***********************************************************************
I                                          I
I                                          I
I                                          I
I                                          I
I                                          I
I         ***********************************************************************
I I.......*     DO 600                                                          *
I I       *     I          I = 1,6                                              *
I I       ***********************************************************************
I I                                        I
I I                                        I
I I                                        I
I I       ***********************************************************************
I I       *     IF( ANGTER(I,1) .EQ. 100.0 )                                    *......O
I I       *     * GO TO 600                                                     *      I
I I       ***********************************************************************      I
I I                                        I                                           I
I I                                        I                                           I
I I                                        I                                           I
I I       ***********************************************************************      I
I I       *     CONSBF(I) = BAFFLE( NDUD, ACOS( COSRAD(M)*COS( ANGTER(I,M) ) ), *      I
I I       *    E DUD ) + SPRLOS(I,M1)                                           *      I
I I       *     DUM=BAFFLE(NCNABT(I),ACOS(COSRAD(M1)*COS(ANGTER(I,M1))),CONPHI(I)*      I
I I       *     CALL                                                            *      I
I I       *    S            CORREC                                              *      I
I I       *    S            ( ANGTER(I,M), I, M )                               *      I
I I       ***********************************************************************      I
I I                                        I                                           I
I I                                        O(..........................................O
I I                                        I
I I       ***********************************************************************
I  .......*  600 CONTINUE                                                       *
I         ***********************************************************************
I                                          I
I                                          I
I                                          I
I         ***********************************************************************
I         *     IF( NOSDRA )  CALL                                              *
I         *    S            CORREC                                              *
I         *    S            ( 0.0, 7, M )                                       *
I         *     IF( NOCZRA )  CALL                                              *
I         *    S            CORREC                                              *
I         *    S            ( ANCZAV(M), 8, M )                                 *
I         *     N1 = NUMFRO(M)                                                  *
I         *     M1 = 1                                                          *
I         ***********************************************************************
I                                          I
```



```
      DO 500
      I          J = 1,N1

      F = FROSIG(J,M)
      SIGRAY = 0.0

      DO 400
      I          I = 1,6

      IF( ANGTER(I,1) .NE. 100.0 )
     E SIGRAY = SIGRAY + ALOGIN( OSPECT( NCNABI(I), CONPHI(I) ) +
     E CONSBF(I) - FRQLCN(J,M)*PATHLN(I) - FRQIDB(J,M)*BOTLOS(I) )
     E *SINXOX( F, I )

  400 CONTINUE

      HK = RANGEH*FRQLCN(J,M)
C
C     DETERMINE SURFACE DUCT CONSTANT BEING USED
C

      IF( NCONSD .EQ. 0 )
     * GO TO 390

      IF( NCONSD .NE. 4 )
     * GO TO 330

      IF( ANGTER(1,1) + ANGTER(2,1) .NE. 200.0 )
     * GO TO 390

  330 CONTINUE
      HSD1 = 2.0*A1 + HK + 60.0
      FTHIRD = FROSIG(J,M)**0.33333333
      G2SD = HZSD/4.0*AMAX1( 2.0, FTHIRD )
      HSD2 = HSD1 + AMAX1( 0.0, FTHIRD*( CONST2 + 5.0*RANGEH ) )

      IF( NZONE .EQ. 1 )
     * GO TO 360
```

```
                                    I
****************************************************************
*       IF( NZONE .EQ. 2 )                                     *.............0
*     * GO TO 340                                              *
****************************************************************
                                    I
                                    I
                                    I
****************************************************************
*       CONST4 = G2SD + A6                                     *
*       HSD4 = HSD1 - A1 + SCSD*RANGEH + CONST4                *
*       HSD = AMIN1( HSD4, HSD2 )                              *
****************************************************************
                                    I
                                    I
                                    I
****************************************************************
*        GO TO 380                                             *...................0
****************************************************************


****************************************************************
*C                                                             *
****************************************************************
                                    I
                                    0(...................................0
                                    I
****************************************************************
*  340 CONTINUE                                                *
*       HSD3 = HSD1 + 2.0*( RSD - RSD1 )*( G2SD - G1SD ) + G1SD *
****************************************************************
                                    I
                                    I
                                    I
                                    I
                                    I
****************************************************************
*       IF( NCONSD .EQ. 4 )                                    *.............0
*     * GO TO 350                                              *
****************************************************************
                                    I
                                    I
                                    I
****************************************************************
*       HSD = HSD3                                             *
****************************************************************
                                    I
                                    I
                                    I
****************************************************************
*        GO TO 380                                             *...................)V
****************************************************************


****************************************************************
*C                                                             *
****************************************************************
                                    I
                                    0(...................................0
                                    I
****************************************************************
*  350 CONTINUE                                                *
*       HSD = AMIN1( HSD3, HSD2 )                              *
****************************************************************
                                    I
                                    I
                                    I
****************************************************************
*        GO TO 380                                             *...................)V
****************************************************************


****************************************************************
*C                                                             *
****************************************************************
                                    I
                                    0(...........................0
                                    I
****************************************************************
*  360 CONTINUE                                                *
*       HSD1 = HSD1 + G1SD*RSD/RSD1                            *
****************************************************************
                                    I
                                    I
                                    I
****************************************************************
*       IF( NCONSD .EQ. 1 )                                    *.......0
*     * GO TO 370                                              *
****************************************************************
                                    I
```

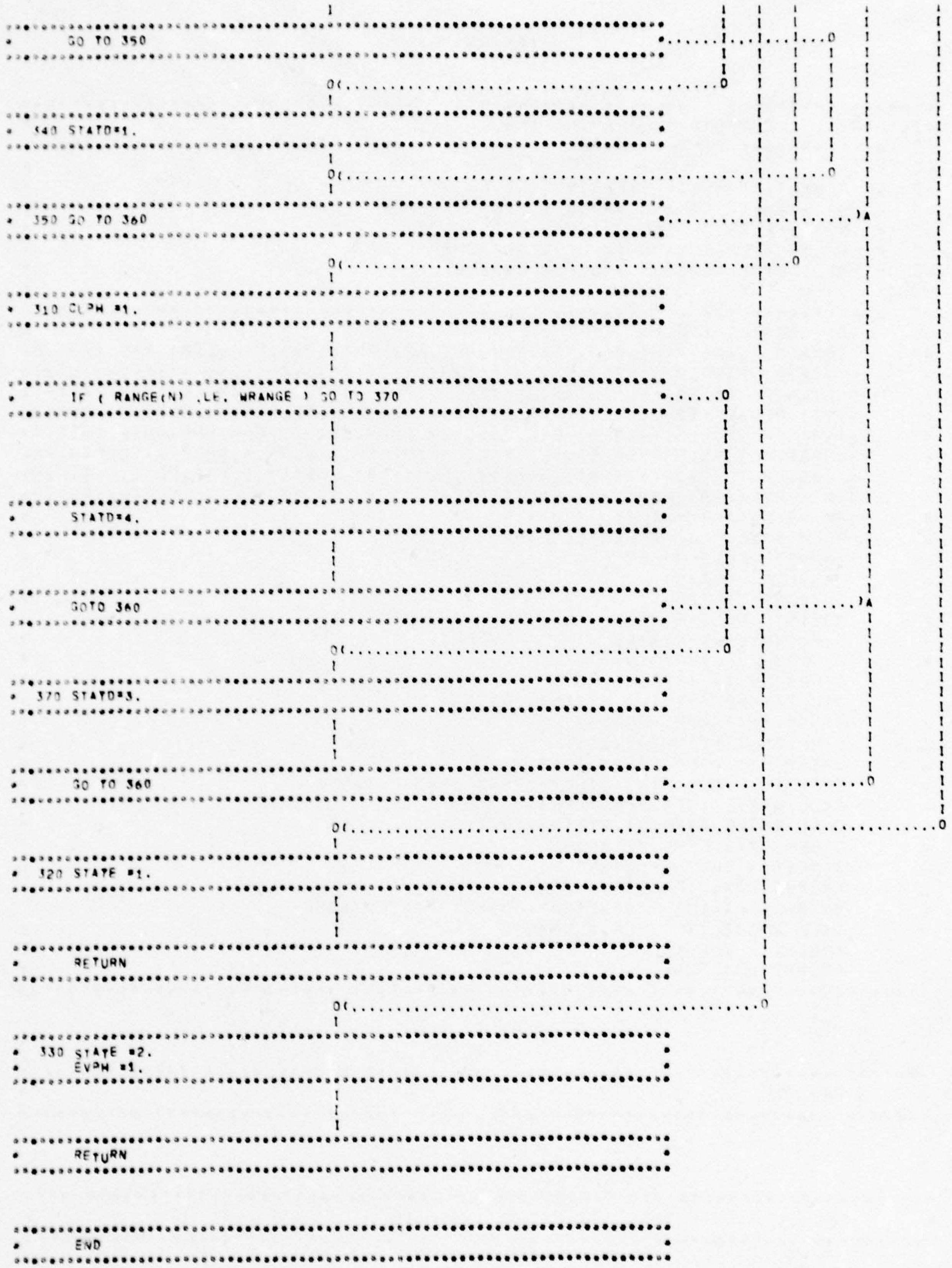
```
      HSD = AMIN1( HSD1, HSD2 )

       GO TO 380

C

  370 CONTINUE
      HSD = HSD1

  380 CONTINUE
      IF( NOSDRA )  SIGSD = ALOGIN( OSPECT( NCNASD, CONASD ) + BFOFSD -
     * HSD
     F )
     F =SINXOX( F, 7 )

  390 CONTINUE
      IF( NOCZRA )SIGCZ=ALOGIN( OSPECT( NCNACZ, CONACZ ) + BFOFCZ - HK )
     F =SINXOX( F, 8 )
      POWSIG(J,M) = ( SIGRAY + SIGSD + SIGCZ )*FRQTRN(J,M)

  300 CONTINUE

      M = 1
      N1= 1
      AFRAC = 0.0
      M1 = NUMFRQ(M)

      DO 100
      I             J = 1,M1

      F = FRQNOS(J,M)
      PTS(J) = POWSIG(J,M)

      IF( NUDIAN(M) .EQ. 1 )
     * GO TO 90

      N1= MORPNT( ANGDGA(M), ANDISO(1,M), NUDIAN(M) ) - 1
      AFRAC = ( ANGDGA(M) - ANDISO(N1,M) )/DLDIAN(N1,M)

   90 CONTINUE
      DI = ALOGIN(DIRSON(J,N1,M)*( 1.0 - AFRAC )+AFRAC*DIRSON(J,N1+1,M))
      POWNOS(J,M) = FRQNCN(J,M)/DI
      PNS(J) = POWNOS(J,M)
```

```
I                                        I
I                     .................................................................................
.....................  100 CONTINUE                                                                   .
                      .................................................................................
                                         I
                                         I
                                         I
                      .................................................................................
                      .       M = 2                                                                   .
                      .       N1= 1                                                                   .
                      .       AFRAC = 0.0                                                             .
                      .       M1 = NUMERO(M)                                                          .
                      .................................................................................
                                         I
                                         I
                                         I
                                         I
                                         I
                      .................................................................................
I.....................        DO 300                                                                  .
I                     .   I             J = 1,M1                                                      .
I                     .................................................................................
I                                        I
I                                        I
I                                        I
I                     .................................................................................
I                     .       F = FRONOS(J,M)                                                         .
I                     .       PTE(J) = POWSIG(J,M)                                                    .
I                     .................................................................................
I                                        I
I                                        I
I                                        I
I                     .................................................................................
I                     .       IF( NUDIAN(M) .EQ. 1 )                                                  .......O
I                     .     * GO TO 80                                                                .      I
I                     .................................................................................      I
I                                        I                                                                   I
I                                        I                                                                   I
I                                        I                                                                   I
I                     .................................................................................      I
I                     .       N1= MORPNT( ANGDGA(M), ANDISO(1,M), NUDIAN(M) ) - 1                     .      I
I                     .       AFRAC = ( ANGDGA(M) - ANDISO(N1,M) )/DLDIAN(N1,M)                       .      I
I                     .................................................................................      I
I                                        I                                                                   I
I                                        O(..........................................................O
I                                        I
I                     .................................................................................
I                     .   80 CONTINUE                                                                 .
I                     .       DI = ALOGIN(DIRSON(J,N1,M)*( 1.0 - AFRAC )+AFRAC*DIRSON(J,N1+1,M))      .
I                     .       POWNOS(J,M) = FRONCN(J,M)/DI                                            .
I                     .       PNE(J) = POWNOS(J,M)                                                    .
I                     .................................................................................
I                                        I
I                                        I
I                                        I
I                     .................................................................................
.....................  300 CONTINUE                                                                   .
                      .................................................................................
                                         I
                                         I
                                         I
                      .................................................................................
                      .       RETURN                                                                  .
                      .................................................................................


                      .................................................................................
                      . 1000 FORMAT(                                                                  .
                      .     F / 42H THE TWO SHIPS ARE IN THE CONVERGENT ZONE   )                      .
                      .C                                                                              .
                      .................................................................................
                                         I
                                         I
                                         I
                                         I
                                         I
                      .................................................................................
                      .       END                                                                     .
                      .................................................................................
```

```
                                (ENTRANCE)
                                    I
                                    I
**************************************************************************
*CREDUCE00        FUNCTION   REDUCE                                      *
*CFREDU000        FUNCTION FOR FORCING RAD POSITIVE                      *
*      FUNCTION   REDUCE                                                 *
*     F ( DUMMY1, DUMMY2, DUMMY3 )                                       *
*C                                                                       *
*C     *******************************************************************
*C     *                                                                **
*C     THIS COMPUTE THE REQUIRED A-CONSTANT SUCH THAT RAD IS POSITIVE    *
*C     *                                                                **
*C     *******************************************************************
*C                                                                       *
*      DUMMY3 = DUMMY1/DUMMY2                                            *
**************************************************************************
                                    I
                                    O(..........................................O
                                    I                                           I
**************************************************************************      I
*   10 CONTINUE                                                          *      I
*      REDUCE = DUMMY1 - DUMMY2*DUMMY3                                   *      I
*      IF( REDUCE .GE. 0.0 )                                             *      I
*     * RETURN                                                           *      I
*      DUMMY4 = DUMMY3                                                   *      I
*      DUMMY3 = DUMMY3*0.99999999                                        *      I
*      WRITE ( 6, 1000 )                                                 *      I
*     W REDUCE,                                                          *      I
*     W DUMMY4, DUMMY3                                                   *      I
**************************************************************************      I
                                    I                                           I
                                    I                                           I
                                    I                                           I
**************************************************************************      I
*      GO TO 10                                                          *......O
**************************************************************************


**************************************************************************
*C                                                                       *
* 1000 FORMAT(                                                           *
*     F/15H BECAUSE RAD = E17.10                                         *
*     F / 20H A WAS CHANGED FROM E17.10, 4H TO E17.10 )                  *
*C                                                                       *
**************************************************************************
                                    I
                                    I
                                    I
**************************************************************************
*      END                                                               *
**************************************************************************
```

```
                                              (ENTRANCE)
                                                  I
                                                  I
                     ********************************************************************
                     *CSINXOX00       FUNCTION SINXOX                                   *
                     *C               COMPUTES BEAM PATTERN DEGRADATION                 *
                     *      FUNCTION SINXOX( F, K )                                     *
                     *C                                                                 *
                     *C     ************************************************************
                     *C     *                                                          *
                     *C     THIS IS THE ROUTINE USING EULERIAN ANGLES AND TRANSFORMATION *
                     *C     *                                                          *
                     *C     ************************************************************
                     *C                                                                 *
                     *      COMMON                                                      *
                     *     C / BEAMCR /                                                 *
                     *     D             BEMCOR(3,8)                                    *
                     *C                                                                 *
                     *      SINXOX = 1.0                                                *
                     ********************************************************************
                                                  I
                                                  I
                                                  I
                     ********************************************************************
I....................*      DO 100                                                      *
I                    *      I            I = 1,3                                        *
I                    ********************************************************************
I                                                 I
I                                                 I
I                                                 I
I                    ********************************************************************
I                    *      SINXOX = ASNXOX( BEMCOR(I,K)*F )*SINXOX                     *
I                    ********************************************************************
I                                                 I
I                                                 I
I                                                 I
I                    ********************************************************************
.....................*  100 CONTINUE                                                    *
                     ********************************************************************
                                                  I
                                                  I
                                                  I
                     ********************************************************************
                     *      RETURN                                                      *
                     ********************************************************************


                     ********************************************************************
                     *C                                                                 *
                     ********************************************************************
                                                  I
                                                  I
                                                  I
                     ********************************************************************
                     *      END                                                         *
                     ********************************************************************
```

```
                               (ENTRANCE)
                                    I
                                    I
************************************************************************
*CSTATE      IDENTIFY D/E STATE                                        *
*C  ********************************************************************
*C  *                                                                  *
*C  *  IDENTIFY D AND E STATE                                          *
*C  *  STATE * STATE E                                                 *
*C  *  STATD * STATE D                                                 *
*C  *  CLPH*CLOSE PHASE                                                *
*C  *EVPH * EVADE PHASE                                                *
*C  *WRANGE * WEAPON RANGE                                             *
*C  *                                                                  *
*C  ********************************************************************
*     SUBROUTINE STAT                                                  *
*     COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*    1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NS1,*
*    2NE1,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*    3128),PEVADE(128),DIFT1,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*    4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*    5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*    6,F0S,FBNS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*    7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*    8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                 *
************************************************************************
                                    I
                                    I
                                    I
************************************************************************
*     IF(CLPH .EQ.1.) GO TO 300                                        *......0
************************************************************************      I
                                    I                                         I
                                    I                                         I
                                    I                                         I
************************************************************************      I
*     IF (N .EQ. NS1) GO TO 310                                        *......I.......0
************************************************************************      I       I
                                    I                                         I       I
                                    I                                         I       I
                                    I                                         I       I
************************************************************************      I       I
*     STATD * 5.                                                       *      I       I
************************************************************************      I       I
                                    I                                         I       I
                                    O(........................................I.......I.......0
                                    I                                         I       I       I
************************************************************************      I       I       I
* 360 IF (EVPH.EQ.1.) GO TO 320                                        *......I.......I.......I......0
************************************************************************      I       I       I      I
                                    I                                         I       I       I      I
                                    I                                         I       I       I      I
                                    I                                         I       I       I      I
************************************************************************      I       I       I      I
*     IF (N.EQ.NE1) GO TO 330                                          *......I...0   I       I      I
************************************************************************      I   I   I       I      I
                                    I                                         I   I   I       I      I
                                    I                                         I   I   I       I      I
                                    I                                         I   I   I       I      I
                                    I                                         I   I   I       I      I
                                    I                                         I   I   I       I      I
************************************************************************      I   I   I       I      I
*     STATE *3.                                                        *      I   I   I       I      I
************************************************************************      I   I   I       I      I
                                    I                                         I   I   I       I      I
                                    I                                         I   I   I       I      I
                                    I                                         I   I   I       I      I
************************************************************************      I   I   I       I      I
*     RETURN                                                           *      I   I   I       I      I
************************************************************************      I   I   I       I      I
                                    O(........................................0   I   I       I      I
                                    I                                             I   I       I      I
************************************************************************          I   I       I      I
* 300 IF ( RANGE(N).LE.WRANGE ) GO TO 340                              *......0   I   I       I      I
************************************************************************      I   I   I       I      I
                                    I                                         I   I   I       I      I
                                    I                                         I   I   I       I      I
                                    I                                         I   I   I       I      I
************************************************************************      I   I   I       I      I
*     STATD *2.                                                        *      I   I   I       I      I
************************************************************************      I   I   I       I      I
                                    I                                         I   I   I       I      I
                                    I                                         I   I   I       I      I
```



```
      GO TO 350
  340 STATD=1.
  350 GO TO 360
  310 CLPH =1.
      IF ( RANGE(N) .LE. WRANGE ) GO TO 370
      STATD=4.
      GOTO 360
  370 STATD=3.
      GO TO 360
  320 STATE =1.
      RETURN
  330 STATE =2.
      EVPH =1.
      RETURN
      END
```

```
                              (ENTRANCE)
                                  I
                                  I
*****************************************************************************
*CSTEER        COMPUTING STEERING ANGLES                                   *
*C   **********************************************************************
*C   *                                                                      *
*C   *   COMPUTING ARRAY STEERING ANGLES                                    *
*C   *   BETAS=BEARING ANG.WRT.BOW FOR SEARCHER                             *
*C   *   BETAE=BEARING ANG.WRT.BOW FOR EVADER                               *
*C   *   DELTAS=DEPRESSION ANG.FOR SEARCHER                                 *
*C   *   DELTAE=DEPRESSION ANG.FOR EVADER                                   *
*C   *                                                                      *
*C   **********************************************************************
*        SUBROUTINE STEERA                                                  *
*        COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*       1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*       2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*       3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*       4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*       5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*       6,F0S,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*       7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*       8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                   *
*        COMMON /ALPHA/ ABEAM1,ABEAM2                                       *
*        SDS=SIN(DELTAS)                                                    *
*        CDS=COS(DELTAS)                                                    *
*        CDE=COS(DELTAE)                                                    *
*        SDE=SIN(DELTAE)                                                    *
*        DIFPY=PYE(N)-PYS(N)                                                *
*        DIFPZ=PZE(N)-PZS(N)                                                *
*        DIFPX=PXE(N)-PXS(N)                                                *
*        CALL ANGVE (VXS(N),VYS(N),THETHS)                                  *
*        STBS=SIN(THETHS+BETAS)                                             *
*        CTBS=COS(THETHS+BETAS)                                             *
*        XPS=DIFPX*CTBS-DIFPY*STBS                                          *
*        YPS=DIFPX*CDS*STBS+DIFPY*CDS*CTBS+DIFPZ*SDS                        *
*        CALL ANGVE (XPS,YPS,ALPHAS)                                        *
*        CALL ANGVE (VXE(N),VYE(N),THETHE)                                  *
*        CTBE=COS(THETHE+BETAE)                                             *
*        STBE=SIN(THETHE+BETAE)                                             *
*        XPE=-DIFPX*CTBE+DIFPY*STBE                                         *
*        YPE=-DIFPX*CDE*STBE-DIFPY*CDE*CTBE-DIFPZ*SDE                       *
*        CALL ANGVE (XPE,YPE,ALPHAE)                                        *
*        ABEAM1 = ALPHAS                                                    *
*        ABEAM2 = ALPHAE                                                    *
*****************************************************************************
                                  I
                                  I
                                  I
*****************************************************************************
*        RETURN                                                             *
*****************************************************************************


*****************************************************************************
*        END                                                                *
*****************************************************************************
```

```
                                                (ENTRANCE)
                                                     I
                                                     I
***************************************************************************
*CSTRTRA00        SUBROUTINE STRTRA                                       *
*CSSTRA000        SUBROUTINE FOR STARTING RAY TRACINGS                    *
*      SUBROUTINE                                                         *
*     S           STRTRA                                                  *
*     S ( N, DEPEND )                                                     *
*C                                                                        *
*C     ********************************************************************
*C     *                                                                 **
*C     *                                                                 **
*C     ********************************************************************
*C                                                                        *
*      COMMON                                                             *
*     C / CONSTN /                                                        *
*     C           DEPSER,                                                 *
*     C           NCONSK,                                                 *
*     C           SPATSE,                                                 *
*     C           DEPEVA,                                                 *
*     C           NCONSL,                                                 *
*     C           SPATEV                                                  *
*      COMMON                                                             *
*     C / PTHLNG /                                                        *
*     C           A1    ,                                                 *
*     C           BI    ,                                                 *
*     C           CDSORD,                                                 *
*     C           CI    ,                                                 *
*     C           DI    ,                                                 *
*     C           DXDC  ,                                                 *
*     C           DZ1   ,                                                 *
*     C           CV    ,                                                 *
*     C           K     ,                                                 *
*     C           PL    ,                                                 *
*     C           SM    ,                                                 *
*     C           V     ,                                                 *
*     C           X     ,                                                 *
*     C           Y1    ,                                                 *
*     C           Y2    ,                                                 *
*     C           TIMCON                                                  *
*      COMMON                                                             *
*     C / RANGES /                                                        *
*     C           NUMANG,                                                 *
*     C           ANGMAX,                                                 *
*     C           DELANG,                                                 *
*     C           DELRAD,                                                 *
*     D           ANGINT(200)                                             *
*     D           RNGMOD(6,200)                                           *
*       COMMON                                                              *
*      C / RAYPAR /                                                         *
*      C               RANGEH,                                              *
*      D               BOTANG(6)         ,                                  *
*      D               DRDXDC(6)         ,                                  *
*      D               PATHLN(6)         ,                                  *
*      D               RANGEC(6)         ,                                  *
*      D               SPI(6)            ,                                  *
*      D               SPT(6)            ,                                  *
*      D               TIR(6)            ,                                  *
*      D               TTR    (6)                                           *
*       COMMON                                                              *
*      C / RAYTRA /                                                         *
*      C               NCONCI,                                              *
*      C               INITLK,                                              *
*      C               Z1    ,                                              *
*      C               Z2    ,                                              *
*      C               SPVRSQ,                                              *
*      C               ANGSTR,                                              *
*      C               ANGARR,                                              *
*      C               ANGBTM,                                              *
*      C               ANGSUR,                                              *
*      C               SPDVER,                                              *
*      C               RANGET                                               *
*C                                                                          *
*C                                                                          *
*       ANGSTR = TIR(N)                                                     *
*       SPDVER = SPATSE/COS( ANGSTR )                                       *
*       SPVRSQ = 1.0/SPDVER/SPDVER                                          *
*       DXDC = 0.0                                                          *
*       PL = 0.0                                                            *
*       RANGET = 0.0                                                        *
*       TIMCON = 0.0                                                        *
*       CALL                                                                *
*      S               LENGTH                                               *
*      S ( N, NCONSK, DEPSER, DEPEND )                                      *
*   900 CONTINUE                                                            *
*****************************************************************************
                                     I
                                     I
                                     I
*****************************************************************************
*       RETURN                                                              *
*****************************************************************************


*****************************************************************************
*C                                                                          *
*****************************************************************************
                                     I
                                     I
                                     I
*****************************************************************************
*       END                                                                 *
*****************************************************************************
```

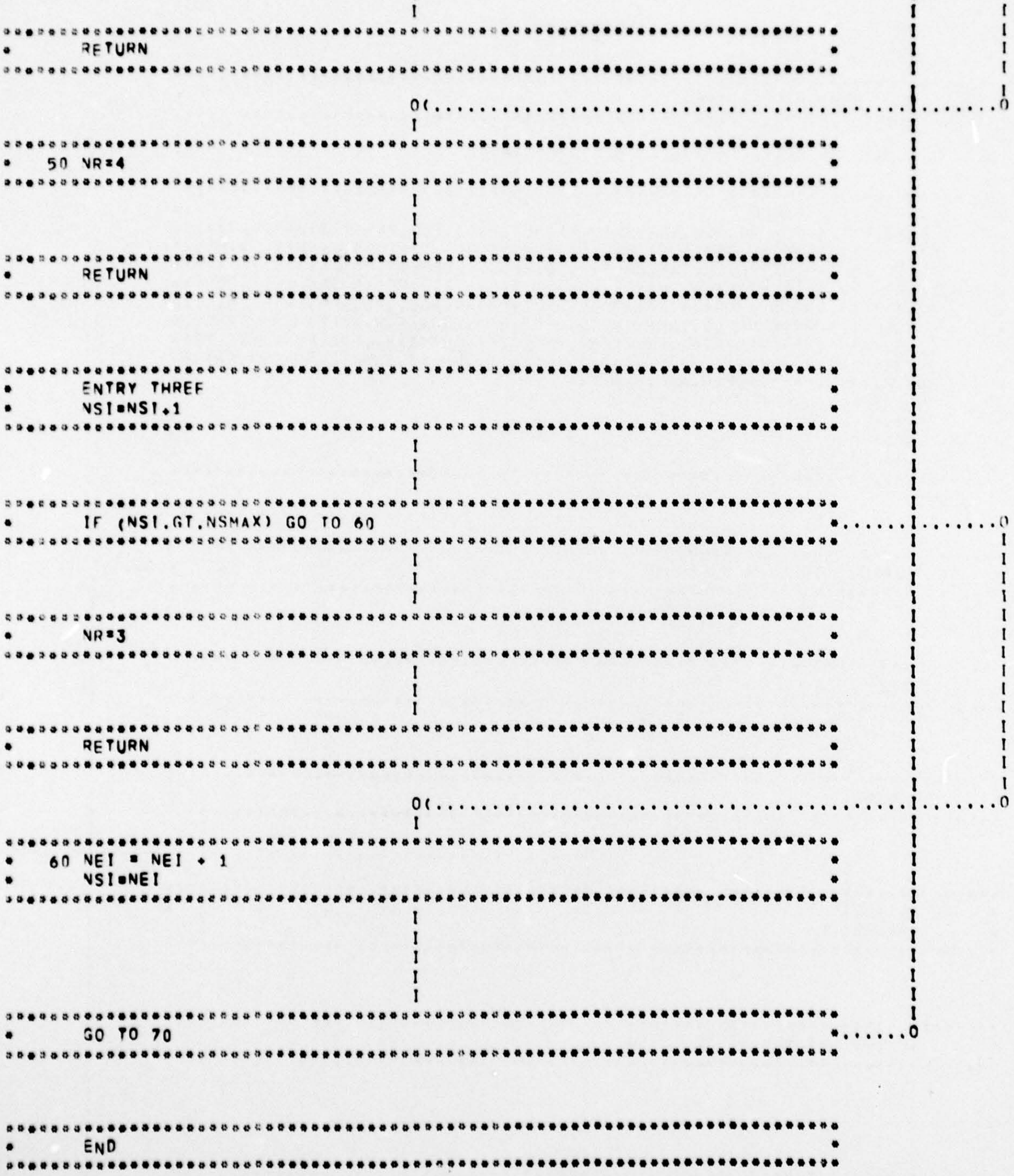
```
                                     (ENTRANCE)
                                          I
                                          I
*************************************************************************
*CTAB_E00    D/E STATE TABLE                                            *
*C   ****************************************************************** *
*C   *                                                                  *
*C   *  D STATE -E STATE TABLE                                          *
*C   *                                                                  *
*C   ****************************************************************** *
*       SUBROUTINE TABLE                                                *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NE1,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),ABC(5*
*      4),DEF(5),PGE(3),GHI(3),JKL(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSF,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,FOS,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBWE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                *
*       COMMON                                                          *
*      C / SIGNAL /                                                     *
*      D               PRYSEV(128)       ,                              *
*      D               PRYSSE(128)       ,                              *
*      D               PRNOEV(128)       ,                              *
*      D               PRNOSE(128)       ,                              *
*      D               PROEVA(128)       ,                              *
*      D               PROSER(128)       ,                              *
*      D               VAREVA(128)       ,                              *
*      D               VARSER(128)       ,                              *
*      D               GMUEVA(128)       ,                              *
*      D               GMUSER(128)       ,                              *
*      D               DEVAEV(128)       ,                              *
*      D               DEVASE(128)       ,                              *
*      D               DEMUEV(128)       ,                              *
*      D               DEMUSE(128)       ,                              *
*      C               THREVA,                                          *
*      C               THRSER,                                          *
*      C               NTIMEN                                           *
*    10 ONE=1.-PROSER(N)                                                *
*       TWO=1.-PROEVA(N)                                                *
*       THREE=1.-PROSER(N)*PRK                                          *
*       FOUR = PROSER(N)                                                *
*       FIVE = PROEVA(N)                                                *
*C                                                                      *
*       PDE(1) = FIVE                                                   *
*       PDS(1) = FOUR                                                   *
*       PRNOEV(1) = TWO                                                 *
*       PRNOSE(1) = ONE/THREE                                           *
*       PRYSEV(1) = FIVE                                                *
*       PRYSSE(1) = FOUR*( 1.0 - PRK )/THREE                            *
*C                                                                      *
*       PDS(2) = FOUR                                                   *
*       PGE(2) = FIVE                                                   *
*       PRNOSE(2)  =  1.0 - PROSER(N)                                   *
*       PRYSSE(2) = FOUR                                                *
*C                                                                      *
*       PGE(3) = TWO                                                    *
*       PGS(3) = FOUR                                                   *
*C                                                                      *
*       PGS(4) = FOUR                                                   *
*C                                                                      *
*       PGS(5)=ONE                                                      *
*************************************************************************
                                          I
                                          I
                                          I
*************************************************************************
*       RETURN                                                          *
*************************************************************************


*************************************************************************
*C                                                                      *
*************************************************************************
                                          I
                                          I
                                          I
*************************************************************************
*       END                                                             *
*************************************************************************
```

```
                           (ENTRANCE)
                               I
                               I
***********************************************************
*CTNLG1000         FUNCTION   TNLG10                      *
*C                 CONVERTS DUMMY VALUE TO DB             *
*      FUNCTION    TNLG10( DUMMY1 )                       *
*C                                                        *
*C     ***************************************************
*C     *                                                 **
*C     COMPUTES DB VALUE USING ALGORITHM OF 10*LOG( VALUE )*
*C     *                                                 **
*C     ***************************************************
*C                                                        *
***********************************************************
                               I
                               I
                               I
***********************************************************
*     IF( DUMMY1 .LE. 1.0E-35 )                            *......O
*     * GO TO 10                                           *      I
***********************************************************       I
                               I                                   I
                               I                                   I
                               I                                   I
***********************************************************       I
*     IF( DUMMY1 .GE. 1.0E+35 )                            *......I......O
*     * GO TO 20                                           *      I      I
***********************************************************       I      I
                               I                                   I      I
                               I                                   I      I
                               I                                   I      I
***********************************************************       I      I
*     TNLG10 = 10.0*ALOG10( DUMMY1 )                      *       I      I
***********************************************************       I      I
                               I                                   I      I
                               I                                   I      I
                               I                                   I      I
***********************************************************       I      I
*     RETURN                                               *       I      I
***********************************************************       I      I
                                                                   I      I
                                                                   I      I
***********************************************************       I      I
*C                                                         *       I      I
***********************************************************       I      I
                               I                                   I      I
                               O(..................................O      I
                               I                                          I
***********************************************************              I
*  10 CONTINUE                                             *              I
*     TNLG10 = -350.0                                      *              I
***********************************************************              I
                               I                                          I
                               I                                          I
                               I                                          I
```

```
                                   I                                                   I
                                   I                                                   I
************************************************************************               I
*     RETURN                                                           *               I
************************************************************************               I
                                                                                       I
                                                                                       I
************************************************************************               I
*C                                                                     *               I
************************************************************************               I
                                   I                                                   I
                                   0(..................................................0
                                   I
************************************************************************
*  20 CONTINUE                                                         *
*     TNLG10 = +350.0                                                  *
************************************************************************
                                   I
                                   I
                                   I
************************************************************************
*     RETURN                                                           *
************************************************************************


************************************************************************
*C                                                                     *
************************************************************************
                                   I
                                   I
                                   I
************************************************************************
*     END                                                              *
************************************************************************
```

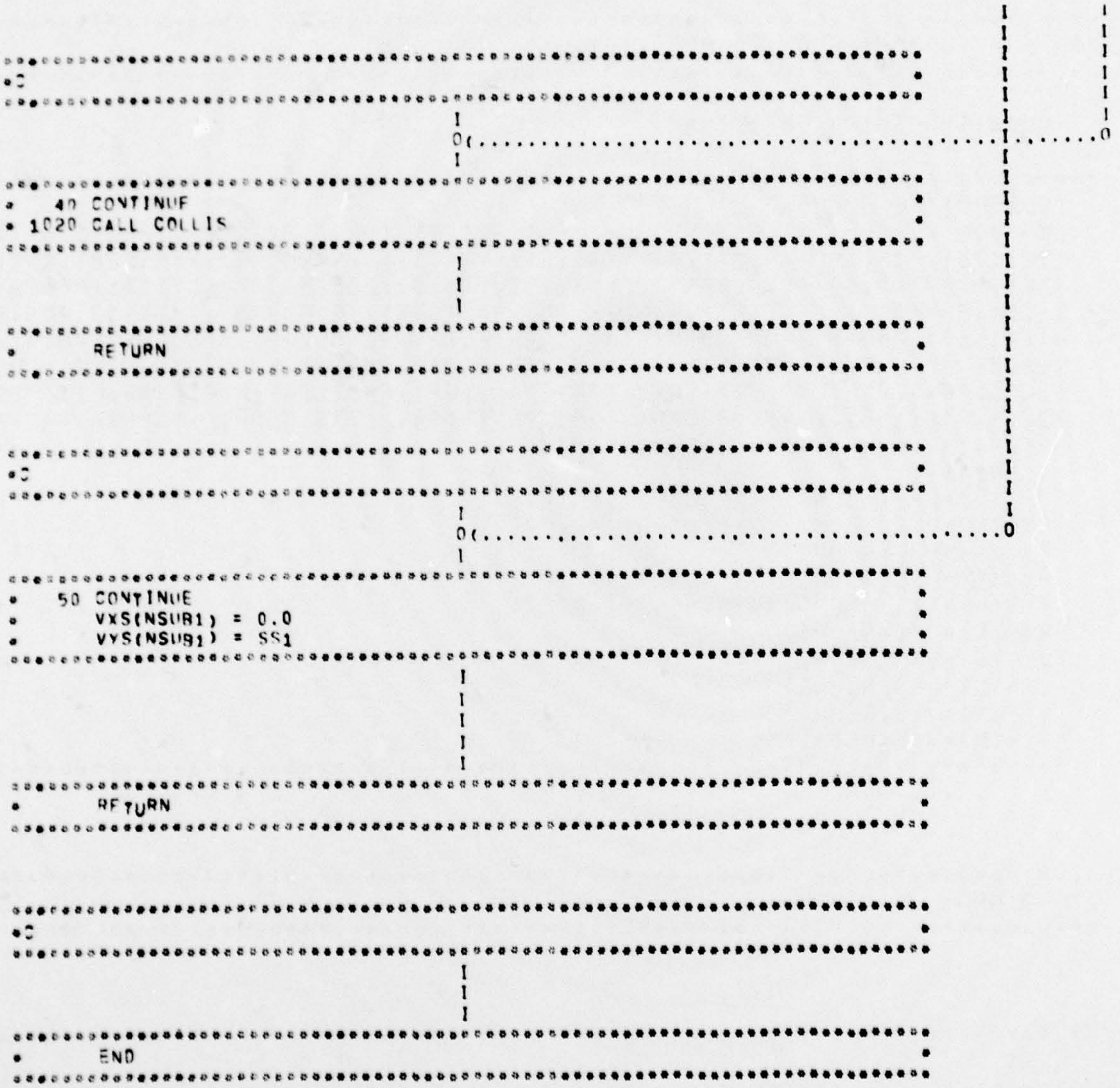
```
                           (ENTRANCE)
                              I
                              I
**************************************************************
*CTRACER00     SUBROUTINE TRACER                             *
*CERROR000     DEBUG PRINTOUT SHOWING PROGRAM FLOW           *
*     SUBROUTINE                                             *
*    S          TRACER                                       *
*    S ( NNFLAG)                                             *
*C                                                           *
*C    ********************************************************
*C    *                                                     **
*C    *                                                     **
*C    ********************************************************
*C                                                           *
*     WRITE(6,90000)NNFLAG                                   *
**************************************************************
                              I
                              I
                              I
**************************************************************
*     RETURN                                                 *
**************************************************************


**************************************************************
*C                                                           *
*90000 FORMAT(8H FLAG = I4///)                                *
*C                                                           *
**************************************************************
                              I
                              I
                              I
**************************************************************
*     END                                                    *
**************************************************************
```

```
                                      (ENTRANCE)
                                          I
                                          I
************************************************************************
*CTREE        PROBABILITY TREE                                         *
*C  *******************************************************************
*C  *                                                                  *
*C    *GENERATION OF NS,NE TABLES DEF PROB TREE                        *
*C  *                                                                  *
*C  *******************************************************************
*     SUBROUTINE PTREE                                                 *
*     COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*    1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*    2NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*    3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*    4),PIS(5),PGE(3),POF(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*    5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*    6,F0S,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*    7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*    8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                                 *
*     NEI=1                                                            *
*     NSI=1                                                            *
*     ENTRY TWO                                                        *
*     NEI=NEI+1                                                        *
************************************************************************
                                          I
                                          O(...........................O
                                          I                            I
************************************************************************     I
*  70 IF(NEI.GT.NEMAX) GO TO 40                                        *.....I.......O
************************************************************************     I       I
                                          I                            I       I
                                          I                            I       I
                                          I                            I       I
************************************************************************     I       I
*     NR=2                                                             *     I       I
************************************************************************     I       I
                                          I                            I       I
                                          I                            I       I
                                          I                            I       I
************************************************************************     I       I
*     RETURN                                                           *     I       I
************************************************************************     I       I
                                                                       I       I
                                          O(...........................I.......O
                                          I                            I
************************************************************************     I
*  40 NEI=NSI                                                          *     I
*     NSI=NSI+1                                                        *     I
************************************************************************     I
                                          I                            I
                                          I                            I
                                          I                            I
************************************************************************     I
*     IF (NSI.GT.NSMAX) GO TO 50                                       *.....I.......O
************************************************************************     I       I
                                          I                            I       I
                                          I                            I       I
                                          I                            I       I
                                          I                            I       I
                                          I                            I       I
************************************************************************     I       I
*     NR=3                                                             *     I       I
************************************************************************     I       I
                                          I                            I       I
```



```
       RETURN

   50 NR=4

       RETURN

       ENTRY THREE
       NSI=NSI+1

       IF (NSI.GT.NSMAX) GO TO 60

       NR=3

       RETURN

   60 NEI = NEI + 1
       NSI=NEI

       GO TO 70

       END
```

```
                                  (ENTRANCE)
                                      !
                                      I
*****************************************************************
*CUPDAT       UPDATE (MAIN SUB)                                 *
*C   ************************************************************
*C   *                                                          *
*C   *  UPDATING SHIP MOTION TO  NEXT TIME INTERVAL             *
*C   *  MECO = ESCAPE  COURSE OPTIONS                           *
*C   *  NPCO =PURSUIT  COURSE  OPTIONS                          *
*C   *  MECO =1 IMPLIES  INVERSE  B RIDER                       *
*C   *  MECO =2 IMPLIES  NTSE                                   *
*C   *  MECO =3 IMPLIES  SAME COURSE                            *
*C   *  NPCO =1 IMPLIES   B RIDER                               *
*C   *  NPCO =2 IMPLIES  COLLIS                                 *
*C   * NPCO =3 IMPLIES SAME  COURSE                             *
*C   *                                                          *
*C   ************************************************************
*       SUBROUTINE UPDATE                                       *
*       COMMON /LABEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128),P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,32,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,STNPSE,STNPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,F0S,FRWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),F0E,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                               *
*       COMMON / INTSPD / VXEINT, VYEINT                                *
*C                                                                      *
*C      VXEINT     = INITIAL EVADER SPEED IN X-DIRECTION           (K*
*C      VYEINT     = INITIAL EVADER SPEED IN Y-DIRECTION           (K*
*C                                                                      *
*       NSUB1=N-1                                                       *
*       DIFPY=PYE(NSUB1)-PYS(NSUB1)                                     *
*       DIFPX=PXE(NSUB1)-PXS(NSUB1)                                     *
*       PESMAG=SQRT(DIFPX**2+DIFPY**2)                                  *
*       ALPXN=(PXE(NSUB1)-PXS(NSUB1))/PESMAG                            *
*       ALPYN=(PYE(NSUB1)-PYS(NSUB1))/PESMAG                            *
*****************************************************************
                                      I
                                      I
                                      I
*****************************************************************
*       IF( (EVPH .EQ. 0.0) .OR. (MECO .EQ. 3) )                *......0
*     * GO TO 20                                                *      I
*****************************************************************      I
                                      I                                I
                                      I                                I
                                      I                                I
*****************************************************************      I
*       IF( MECO .EQ. 2 )                                       *......I.......0
*     * GO TO 10                                                *      I       I
*****************************************************************      I       I
                                      I                                I       I
                                      I                                I       I
                                      I                                I       I
                                      I                                I       I
                                      I                                I       I
*****************************************************************      I       I
*  940 CALL INVBRI                                              *      I       I
*****************************************************************      I       I
                                      I                                I       I
                                      I                                I       I
                                      I                                I       I
*****************************************************************      I       I
*        GO TO 30                                               *......I.......I.......0
*****************************************************************      I       I       I
                                                                       I       I       I
                                                                       I       I       I
*****************************************************************      I       I       I
*C                                                              *      I       I       I
*****************************************************************      I       I       I
                                      I                                I       I       I
                                      0(...............................I.......0       I
                                    I                                           I               I
*************************************************************************       I               I
*   10 CONTINUE                                                         *       I               I
*  990 CALL NTSE                                                        *       I               I
*************************************************************************       I               I
                                    I                                           I               I
                                    I                                           I               I
                                    I                                           I               I
*************************************************************************       I               I
*       GO TO 30                                                        *.......I..............)V
*************************************************************************       I               I
                                                                                I               I
                                                                                I               I
*************************************************************************       I               I
*C                                                                      *       I               I
*************************************************************************       I               I
                                    I                                           I               I
                                    O(..........................................O               I
                                    I                                                           I
*************************************************************************                       I
*   20 CONTINUE                                                         *                       I
*      VXE(NSUB1) = VXEINT                                              *                       I
*      VYE(NSUB1) = VYEINT                                              *                       I
*************************************************************************                       I
                                    I                                                           I
                                    O(..........................................................O
                                    I
*************************************************************************
*   30 CONTINUE                                                         *
*************************************************************************
                                    I
                                    I
                                    I
                                    I
*************************************************************************
*      IF( (CLPH .EQ. 0.0) .OR. (NPCO .EQ. 3) )                         *.......O
*     * GO TO 50                                                        *       I
*************************************************************************       I
                                    I                                           I
                                    I                                           I
                                    I                                           I
*************************************************************************       I
*      IF( NPCO .EQ. 2 )                                                *.......I.......O
*     * GO TO 40                                                        *       I       I
*************************************************************************       I       I
                                    I                                           I       I
                                    I                                           I       I
                                    I                                           I       I
*************************************************************************       I       I
* 1000 CALL BRIDER                                                      *       I       I
*************************************************************************       I       I
                                    I                                           I       I
                                    I                                           I       I
                                    I                                           I       I
*************************************************************************       I       I
*      RETURN                                                           *       I       I
*************************************************************************       I       I
                                                                                I       I
```

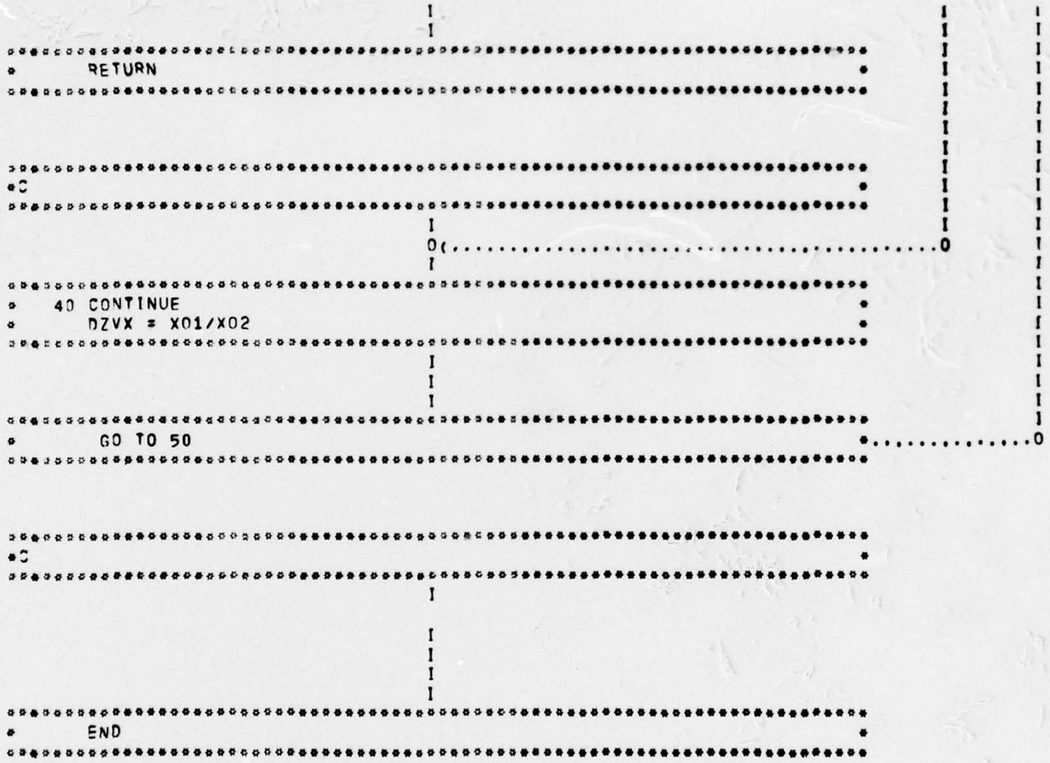


```
C

   40 CONTINUE
 1020 CALL COLLIS

      RETURN

C

   50 CONTINUE
      VXS(NSUB1) = 0.0
      VYS(NSUB1) = SS1

      RETURN

C

      END
```

```
                                (ENTRANCE)
                                    I
                                    I
********************************************************
*CUPPOS       UPPOS -UPDATES POSITIONS                 *
*C   ***************************************************
*C   *                                                 *
*C     *UPDATING SHIPS POSITION                        *
*C   *                                                 *
*C   ***************************************************
*       SUBROUTINE UPPOS                               *
*       COMMON /LAPEL/ RI,RCJ,SS1,SE1,HS1,HE1,PXS(128),PXE(128),PYS(128);P*
*      1YE(128),PZS(128),PZE(128),VXS(128),VYS(128),VXE(128),VYE(128),NSI,*
*      2NEI,N,BETAS,BETAE,DELTAS,DELTAE,B2,PDS(5),PDE(3),PKILL(128),PPATH(*
*      3128),PEVADE(128),DIFTI,RANGE(128),STATD,STATE,PGS(5),PKDS(5),POS(5*
*      4),PIS(5),PGE(3),POE(3),PIE(3),CLPH,EVPH,WRANGE,BPS,BPE,PHIE,PHIS,A*
*      5LSUBE,ALSUBS,SINPSE,SINPEV,MECO,NPCO,BSP,BEP,NR,K,EDEPTH,SDEPTH,RC*
*      6,FOS,FBWS,F1S,PTS(128),FXS(128),PNS(128),FNS(128),FOE,FBSE,F1E,F2E*
*      7,F2S,PTE(128),FXE(128),PNE(128),FNE(128),XE,XS,SUMKIL,SUMEVA,PRE,P*
*      8RK,PE(3),ALPXN,ALPYN,NSMAX,NEMAX                               *
*       NSUB1=N-1                                                      *
*       PXE(N)=PXE(NSUB1)+VXE(NSUB1)*DIFTI                             *
*       PYE(N)=PYE(NSUB1)+VYE(NSUB1)*DIFTI                             *
*       PZE(N)=PZE(NSUB1)                                              *
*       PXS(N)=PXS(NSUB1)+VXS(NSUB1)*DIFTI                             *
*       PYS(N)=PYS(NSUB1)+VYS(NSUB1)*DIFTI                             *
*       PZS(N)=PZS(NSUB1)                                              *
*       VXS(N)=VXS(NSUB1)                                              *
*       VYS(N)=VYS(NSUB1)                                              *
*       VXE(N)=VXE(NSUB1)                                              *
*       VYE(N)=VYE(NSUB1)                                              *
********************************************************************
                                    I
                                    I
                                    I
********************************************************************
*    25 RETURN                                                     *
********************************************************************


********************************************************************
*       END                                                        *
********************************************************************
```

```
                                   (ENTRANCE)
                                       I
                                       I
*************************************************************************
*CVERTEX00           SUBROUTINE VERTEX                                  *
*CSVERT000           SUBROUTINE TO FIND LOWER VERTEX POINT              *
*       SUBROUTINE                                                      *
*      S             VERTEX                                             *
*      S ( I, CU, ZVLO, DPTOZ1, LOOKUP )                                *
*C                                                                      *
*C     ******************************************************************
*C     *                                                               **
*C     THIS FINDS DEPTH AT WHICH RAY VERTEXES                           *
*C     *                                                               **
*C     ******************************************************************
*C                                                                      *
*       COMMON                                                          *
*      C / LCONST /                                                     *
*      C             NULAPO,                                            *
*      C             DEPROT,                                            *
*      D             CONSG0(128)        ,                               *
*      D             CONSG1(128)        ,                               *
*      D             CONSG2(128)        ,                               *
*      D             CONSV0(128)        ,                               *
*      D             DELTAZ(128)        ,                               *
*      D             DEPKYD(128)        ,                               *
*      D             SLOPEJ(128)        ,                               *
*      D             SPDKYD(128)                                        *
*C                                                                      *
*C      CONSG1( )  = G1 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C      CONSG2( )  = G2 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C      CONSV0( )  = V0 CONSTANT AS COMPUTED BY CONTINUOUS DERIVATIVE ROUT*
*C      CU         =                                                    *
*C      DEPKYD( )  = DEPTH (IN K-YD) TO TOP OF LAYER FROM OCEAN SURFACE *
*C      I          = LAYER IN WHICH RAY VERTEXEXES                      *
*C      SPDKYD( )  = SPEED OF SOUND PROPAGATION (IN K-YD/SEC)           *
*C                                                                      *
*       NAMELIST / ERROR / X01, X02, X03, DZVX, DZVX1, DZVX2, CUCAL.    *
*      N DELDEP, VELOCI, DELTOZ, DPTOZ1, K, J                           *
*C                                                                      *
*       X01 = CU - CONSV0(I)                                            *
*       X02 = CONSG0(I) - 2.0*CONSG2(I)*X01                             *
*       X03 = 2.0*( X01*CONSG2(I)*CONSG2(I) - CONSG1(I) )               *
*************************************************************************
                                       I
                                       I
                                       I
*************************************************************************
*       IF( X03 .EQ. 0.0 )                                              *......O
*      * GO TO 40                                                       *      I
*************************************************************************      I
                                       I                                       I
                                       I                                       I
                                       I                                       I
*************************************************************************      I
*       X01 = SQRT( X02*X02 - 2.0*X01*X03 )                             *      I
*       DZVX1 = ( X02 + X01 )/X03                                       *      I
*       DZVX2 = ( X02 - X01 )/X03                                       *      I
*************************************************************************      I
                                       I                                       I
                                       I                                       I
                                       I                                       I
                                       I                                       I
                                       I                                       I
*************************************************************************      I
*       IF( LOOKUP .GT. 0 )                                             *......I.......O
*      * GO TO 10                                                       *      I       I
*************************************************************************      I       I
                                       I                                       I       I
                                       I                                       I       I
                                       I                                       I       I
*************************************************************************      I       I
*       DELDEP = DEPKYD(I+1) - DEPKYD(I)                                *      I       I
*       IF(((DZVX1.LT.DPTOZ1).OR.(DZVX1.LT.0.0)).OR.(DZVX1.GT.DELDEP))  *      I       I
*      * DZVX1 = 1000.0                                                 *      I       I
*       IF(((DZVX2.LT.DPTOZ1).OR.(DZVX2.LT.0.0)).OR.(DZVX2.GT.DELDEP))  *      I       I
*      * DZVX2 = 1000.0                                                 *      I       I
*       DZVX = AMIN1( DZVX1, DZVX2 )                                    *      I       I
*************************************************************************      I       I
                                       I                                       I       I
```

```
                                    I                                           I       I
                                    I                                           I       I
**************************************************************************      I       I
*      IF( DZVX .NE. 1000.0 )                                            *......I.......I......O
*       * GO TO 20                                                       *      I       I      I
**************************************************************************      I       I      I
                                    I                                           I       I      I
                                    O(..........................................I.......I......I.......O
                                    I                                           I       I      I       I
**************************************************************************      I       I      I       I
*    30 CONTINUE                                                         *      I       I      I       I
*       K = LOOKUP                                                       *      I       I      I       I
*       J = I                                                            *      I       I      I       I
*       DELDEP = DELTAZ(I)                                               *      I       I      I       I
*       VELOCI = CU                                                      *      I       I      I       I
*       DELTOZ = DPTOZ1                                                  *      I       I      I       I
*       WRITE( 6, ERROR )                                                *      I       I      I       I
*       CALL DUMP                                                        *      I       I      I       I
*C                                                                       *      I       I      I       I
**************************************************************************      I       I      I       I
                                    I                                           I       I      I       I
                                    O(..........................................I.......O      I       I
                                    I                                           I              I       I
**************************************************************************      I              I       I
*    10 CONTINUE                                                         *      I              I       I
*       IF( (DZVX1 .GT. DPTOZ1) .OR. (DZVX1 .LT. 0.0) )  DZVX1 = -1000.0 *      I              I       I
*       IF( (DZVX2 .GT. DPTOZ1) .OR. (DZVX2 .LT. 0.0) )  DZVX2 = -1000.0 *      I              I       I
*       DZVX = AMAX1( DZVX1, DZVX2 )                                     *      I              I       I
**************************************************************************      I              I       I
                                    I                                           I              I       I
                                    I                                           I              I       I
                                    I                                           I              I       I
**************************************************************************      I              I       I
*       IF( DZVX .EQ. (-1000.0) )                                        *......I..............I......)A
*       * GO TO 30                                                       *      I              I       I
**************************************************************************      I              I       I
                                    I                                           I              I       I
                                    I                                           I              I       I
                                    I                                           I              I       I
                                    O(..........................................I..............O       I
                                    I                                           I                      I
**************************************************************************      I                      I
*    20 CONTINUE                                                         *      I                      I
*       CUCAL = 1.0/FVELOC( DZVX, I )**2                                 *      I                      I
**************************************************************************      I                      I
                                    I                                           I                      I
                                    I                                           I                      I
                                    I                                           I                      I
**************************************************************************      I                      I
*       IF( ABS( 1.0 - CUCAL/CU ) .GT. 1.0E-5 )                          *......I......................O
*       * GO TO 30                                                       *      I
**************************************************************************      I
                                    I                                           I
                                    O(..........................................I.......O
                                    I                                           I       I
**************************************************************************      I       I
*    50 CONTINUE                                                         *      I       I
*       ZVLO = DEPKYD(I) + DZVX                                          *      I       I
*   900 CONTINUE                                                         *      I       I
**************************************************************************      I       I
                                    I                                           I       I
```



```
      RETURN
C
   40 CONTINUE
      DZVX = X01/X02
      GO TO 50
C
      END
```